NPS72-87-001CR

NAVAL POSTGRADUATE SCHOOL

Monterey, California

CONTRACTOR REPORT

NUMERICAL COMPUTATION OF RING-SYMMETRIC

SPACECRAFT EXHAUST PLUMES

by

Joseph Falcovitz

January 1987

Approved for public release; distribution unlimited.

Prepared for: Strategic Defense Initiative Office
The Pentagon
Washington, DC 20301-7100

NAVAL POSTGRADUATE SCHOOL
Monterey, California

RADM R. C. Austin
Superintendent

D. A. Schrady
Provost

The work reported herein was performed for the Naval Postgraduate School by Dr. Joseph Falcovitz under contract N62271-86-M-0214. The work presented in this report is in support of "Rarefied Gas Dynamics of Laser Exhaust Plume" sponsored by the Strategic Defense Initiative Office/Directed Energy Office. This is a partial report for that contract. The work provides information concerning numerical computation of the flow in spacecraft exhaust plumes. The project at the Naval Postgraduate School is under the cognizance of Distinguished Professor A. E. Fuhs who is principal investigator.

Prepared by:

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS NONE
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NPS72-87-001CR	5. MONITORING ORGANIZATION REPORT NUMBER(S) NPS72-87-001CR
6a. NAME OF PERFORMING ORGANIZATION JOSEPH FALCOVITZ 6b. OFFICE SYMBOL (If applicable) 72	7a. NAME OF MONITORING ORGANIZATION NAVAL POSTGRADUATE SCHOOL, CODE 72
6c. ADDRESS (City, State, and ZIP Code) Research Contractor Naval Postgraduate School, Code 72 Monterey, CA 93943-5100	7b. ADDRESS (City, State, and ZIP Code) Space Systems Academic Group Monterey, CA 93943-5100
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Strategic Defense Initiative Office 8b. OFFICE SYMBOL (If applicable) SDIO/DEO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MIPR DGAA60045
8c. ADDRESS (City, State, and ZIP Code) SDIO/DEO Washington, DC 20301-7100	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. PE63221 | PROJECT NO. | TASK NO | WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

Numerical Computation of Ring-Symmetric Spacecraft Exhaust Plumes

12. PERSONAL AUTHOR(S)
JOSEPH FALCOVITZ

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Contractor Report	FROM Jan 86 TO Aug 86	January 1987	54

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Laser Exhaust, Exhaust Plume, Method of Characteristics, Inverse Marching, Ring Plumes

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report supplements report NPS72-86-003CR. It provides further details about the code JET and the numerical schemes on which it is based: inverse marching characteristic and semi-inverse marching characteristic (SIMA) schemes. The computational procedure is described in some detail. The principles of operation of the code JET are outlined, including a glossary of all major arrays, variables and subroutines. Finally, the full listing of the JET code is reproduced.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL ALLEN E. FUHS, Distinguished Professor	22b. TELEPHONE (Include Area Code) (408)646-2948 | 22c. OFFICE SYMBOL 72

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—606-24.

ABSTRACT

This report supplements report NPS72-86-003CR. It provides further details about the code JET and the numerical schemes on which it is based : inverse marching characteristic and semi-inverse marching characteristic (SIMA) schemes. The computational procedure is described in some detail. The principles of operation of the code JET are outlined, including a glossary of all major arrays, variables and subroutines. Finally, the full listing of the JET code is reproduced.

ACKNOWLEDGEMENT

This work was conducted as part of a laser exhaust study under the cognizance of Distinguished Professor Allen E. Fuhs. I deeply appreciate and wish to thank Professor Fuhs for his continuous support and guidance.

TABLE OF CONTENTS

NOMENCLATURE **followed by units (if any) and CODE NOTATION (if any)**

a	sound speed (m sec^{-1})
B	breakdown parameter [5,6,7]
$C^{\pm}$	characteristic lines inclined at $(\theta \pm \mu)$
D	molecular diameter (hard spheres) (m)
M	Mach number
n	number density (molecules/m^3)
p	pressure (Pa)
S	coordinate along streamlines (m)
u	flow velocity (m/sec)
x	axial cartesian coordinate
y	radial cartesian coordinate
γ	specific-heat ratio (G)
η	length coordinate along fan characteristics (C^+) (m)
θ	inclination of flow velocity vector
λ_0	mean free path at stagnation conditions (m)
μ	Mach angle ($\sin\mu = 1/M$) (MU)
ν	Prandtl-Meyer function (NU)
ξ	length coordinate along transverse (C^-) characteristic
σ	collision cross-section πD^2 (m^2) (SIGMA)
τ	molecular opacity (expected number of collisions by a fast invading molecule) (XI)
φ	collision frequency (sec^{-1})
ω	symmetry index (0 - planar flow, 1 - axisymmetric flow) (DELTA)
Γ	the fraction $\left[(\gamma+1)/(\gamma-1)\right]^{1/2}$
$(\nu+\theta)$	Riemann invariant along C^- (RM)
$(\nu-\theta)$	Riemann invariant along C^+ (RP)

INDICES

$(\ )_0$	a specific point in the CRW (x_0, y_0) (Also : stagnation conditions)
$(\ )_1$	nozzle exit conditions
$(\ )_L$	limiting CRW characteristic $(p=0)$
$(\ )_f$	final CRW characteristic (boundary of numerical integration)
$(\ )_c$	corner of CRW

EMPTY PAGE

1. INTRODUCTION

In a recent report [1] a mixed numerical/analytical approach to the computation of a ring-symmetric spacecraft exhaust plume was presented. The numerical scheme had been implemented in a code named "JET" which is capable of generating whole-plume flow fields, while the analytic approximation is restricted to the ring-symmetric centered rarefaction waves (CRW) that flank the plume. The present report is intended to serve as a supplement to [1] in providing details on the computational scheme and the code JET.

The spacecraft exhaust flow (Fig. 1 of [1]) is idealized as a ring-symmetric steady isentropic expansion of an ideal gas. The nozzle lips are assumed sharp; the supersonic flow from the exit surface of the ring-nozzle is assumed uniform, and the background is considered to be perfect vacuum.

The standard scheme for computing such idealized ring-plumes is the classical (direct) method of characteristics [2] . At a preliminary phase of the present laser exhaust study, a code AXSYM [3] was written for computing ring-plumes using this method. A notorious shortcoming of the direct method of characteristics is that the solution grid is highly irregular, being formed by the (oblique) intersection of the C^+ and the C^- families of characteristic lines. We first encountered a difficulty with this grid while seeking a scheme for integrating the molecular opacity along a straight line [1] . This computation would have required rather complex coding for the geometry of intersection between a straight line and an irregular grid. It seemed preferable to opt for a computation scheme that would produce a more regular grid, even at the expense of some loss of accuracy. Such scheme is the inverse marching method of characteristics [4] .

Generally the marching in this type of scheme is in the downstream direction, i.e., the y direction in our case. Grid points are located on a succession of constant-y rows, thereby introducing a measure of regularity in the solution grid. Just two rows have to be stored in the computer core memory - the "old" row and the "new" row, whereas in the direct method of characteristics whole grid-image matrices are required to reside simultaneously in core memory.

The first version of the JET code was based on the inverse marching scheme given by Zucrow and Hoffman (Section 12-5 in [4]), where the flow variables were the ***two cartesian velocity components.*** The computation seemed accurate everywhere, except within the centered rarefaction wave (CRW). In an attempt to replicate a planar CRW (Prandtl-Meyer flow), the numerical solution exhibited an

instability : Mach number increased along the (low pressure) boundary characteristic line, rather than remain constant.

A ***qualitative*** explanation for this instability is the following. Flow gradients in a CRW are inversely proportional to distance from the corner, so that the inverse marching scheme gives rise to an amplification of interpolation errors at every marching step, leading to an apparently divergent (unstable) numerical solution. Increasing the order of interpolation from linear to cubic did not eliminate the instability.

Looking for a scheme that would replicate a planar CRW accurately, we tried the modified marching idea as presented by Zucrow and Hoffman for 1-D time-dependent flows (Sections 19-6(a) and 19-6(j) of [4]). In this scheme new grid points are determined by forwardly extending a "primary" family of continuous characteristic lines from old grid points. The primary family in a CRW is the characteristics fanning out from the corner (we assume it is the C^+ family). By choosing this modified scheme, the interpolation for trace points obtained from reversely extended C^+ lines was eliminated. However, the corresponding interpolation for the transverse C^- characteristics remained, and with it the aforementioned instability.

In order to replicate a planar CRW, we had to replace the flow variables by the *Riemann invariants* $(\nu \pm \theta)$. In a C^+ planar CRW, the Riemann invariant $(\nu + \theta)$ is uniformly constant, so that the interpolation in $(\nu + \theta)$ due to reversely extending C^- characteristics introduces no error at all. This scheme, which we named SIMA (Semi Inverse Marching Algorithm), was indeed verified to replicate a planar CRW exactly, when implemented in the code JET.

The plan of this report is the following. In Ch. 2 we supplement the description of the numerical scheme given in Ch. 2 of [1] , by adding more details on the computational procedure. A description of the code JET is given in Ch. 3, and the code listing is reproduced in Ch. 4.

Note on symmetry :
The code JET has two symmetry options. When DELTA = 1 a ring-symmetry is in effect; when DELTA = 0, a planar symmetry is in effect. An axisymmetric jet exiting in the $\mathbf{y}$ direction from the same nozzle aperture along the $\mathbf{x}$ axis can readily be computed by replacing all terms in the code that correspond to $\sin(\theta)/\mathbf{y}$ in the compatibility equations (2.1-1), by $\cos(\theta)/\mathbf{x}$. In that case the coding is virtually unchanged, and the only care that should be exercised is for the difference equations for new grid points on or near the $\mathbf{y}$ axis. Also, all reference to the analytic approximation of the ring-symmetric CRW [1] should be deleted in this case, as it is designed specifically for ring-symmetry.

2. THE COMPUTATIONAL SCHEME

A basic description of the semi inverse (SIMA) and inverse marching schemes was given in Ch. 2 of [1] . We supplement this description by specifying the slightly modified definition of Riemann invariants in the code, and by giving information about some ancillary computations.

2.1 Riemann Invariants

The compatibility equations whose integration constitutes the numerical solution to the governing equations [1] are expressed in terms of the Riemann invariants as follows :

$$\begin{aligned} &\text{Along } C^{+} \quad \ldots\ldots \quad (\nu-\theta)_4 = (\nu-\theta)_2 + \omega \sin\mu_{24} \sin\theta_{24} \Delta\eta / y_{24} \\ &\text{Along } C^{-} \quad \ldots\ldots \quad (\nu+\theta)_4 = (\nu+\theta)_1 + \omega \sin\mu_{14} \sin\theta_{14} \Delta\xi / y_{14} \end{aligned} \tag{2.1-1}$$

The Riemann invariants $(\nu \pm \theta)$ are modified for convenience, by adding a constant to both ν and θ. The new definitions of $\nu(M)$ and θ are :

$$\begin{aligned} &\nu(M) = -\Gamma \arctan(\Gamma q) + \arctan(q) \\ &q = \left(M^2 - 1\right)^{-1/2} \\ &\theta \rightarrow \theta - \theta_L \end{aligned} \tag{2.1-2}$$

Thus, in a Prandtl-Meyer flow with entry Mach number of M_1, the modified values of both $\nu(M)$ and θ vanish as $M \rightarrow \infty$. As a consequence, in a C^{+} Prandtl-Meyer flow the modified invariant $(\nu + \theta)$ vanishes uniformly. In this modified form, the computation of M from $\nu(M)$ is readily done by performing standard Newton-Raphson iterations (in RFUNC), using the derivative :

$$\nu'(q) = -(\Gamma^2 - 1) \left[(1 + \Gamma^2 q^2)(1 + q^2)\right]^{-1} \tag{2.1-3}$$

2.2 The Integration Scheme for a New Grid Point

The integration scheme has been sketched in Ch. 2 of [1]. It is performed in INVMAR for inverse marching points or in SEMINV for semi-inverse marching (SIMA) points. The computational scheme is specified via the following seven-step procedure :

INVMAR (Inverse Marching)

(a) Grid : At this stage the new grid point has already been defined.

(b) Predictor : Flow variables are the interpolated (linear nearest-neighbor) value on the old row for a point having the new grid $\mathbf{x}$ coordinate (x_4).

(c) Centered variables : Denote the Riemann invariants by

$$\begin{aligned} RM &= (\nu + \theta) \\ RP &= (\nu - \theta) \end{aligned} \qquad (2.2\text{-}1)$$

then centered values for segments (1,4) and (2,4) (using code notation) are :

$$\begin{aligned} RM14 &= (RM1 + RM4)/2 \qquad & RP14 &= (RP1 + RP4)/2 \\ RM24 &= (RM2 + RM4)/2 \qquad & RP24 &= (RP2 + RP4)/2 \end{aligned} \qquad (2.2\text{-}2)$$

All other centered flow variables are computed from the centered Riemann invariants by calling RFUNC.

(d) Inverse Extension : old trace points x_1, x_2 are evaluated from the geometrical relations

$$\begin{aligned} &\text{Along } C^- \quad \ldots\ldots \quad y_{new} - y_{old} = (x_4 - x_1)\tan(\theta_{14} - \mu_{14}) \\ &\text{Along } C^+ \quad \ldots\ldots \quad y_{new} - y_{old} = (x_4 - x_2)\tan(\theta_{24} + \mu_{24}) \end{aligned} \qquad (2.2\text{-}3)$$

(e) Interpolation : find Riemann invariants RM, RP at old trace points x_1 and x_2 through nearest-neighbor linear interpolation by calling INTERP.

(f) Integration : Using the compatibility relations in finite-difference form (2.1-1) with segment-centered coefficients, compute iteration-updated values of Riemann invariants at new grid point.

(g) Corrector : if values of Riemann invariants and old trace points x_1, x_2 are not sufficiently convergent, resume the procedure at step (c) above.

SEMINV (Semi Inverse Marching - SIMA)

(a) Grid : New grid point (x_4) is determined as part of the SIMA scheme at step (d) below.

(b) Predictor : Flow variables are those of point (x_2, y_{old}).

(c) Centered variables : Identical to step (c) above.

(d) Semi-Inverse Extension : new grid point x_4 and old trace point x_1 are evaluated from the geometrical relations in Eq. (2.2-3) above.

(e) Interpolation : find Riemann invariants RM, RP at old trace point x_1 through nearest-neighbor linear interpolation by calling INTERP.

(f) Integration : Identical to step (f) above.

(g) Corrector : Identical to step (g) above, except for replacing x_2 in the convergence test by x_4.

2.3 Boundary Conditions

On the vacuum side the boundary conditions ($p=0$) can only be approximately implemented in a method of characteristics scheme. We do so by ending the computation on a certain "final" C^+ fan characteristic line that starts out with a sufficiently high Mach number M_f at the corner (typically $M_f = 34$). The marching computation of new grid points on the boundary C^+ characteristic via the SIMA scheme is identical to that of C^+ characteristics within the ring-symmetric CRW. It is noted that under this boundary scheme some outflow takes place through the boundary characteristic line, so that the total mass flow through a row $y = y_{new}$ decreases slightly as y_{new} increases.

At the nozzle exit the boundary conditions are assumed to be uniform outflow in the radial (y) direction with Mach number M_1. At the nozzle lip, the SIMA integration starts out from a presumed planar CRW (Prandtl-Meyer flow) at the corner (i.e., the associate CRW in the terminology of Ch. 3 in [1]).

At the plane of symmetry ($x=0$) the boundary condition is simply $\theta = \pi/2$. However, this condition is implemented indirectly, by assuming that the flow at virtual grid points with $x < 0$ is a mirror-image of the flow at the corresponding $x > 0$ points. The reason is that when a new grid point of $x_4 = 0$ or of x_4 sufficiently close to zero is considered for inverse-marching integration, the inversely extended trace point (x_1, y_{old}) can be at $x < 0$. Considering the subtraction of θ_L from θ as in Eq.(2.1-2), the reflection rules are :

$$RM \rightarrow RP + (\pi - 2\theta_L)$$

$$RP \rightarrow RM - (\pi - 2\theta_L) \tag{2.3-1}$$

where values on the left and right of the $\rightarrow$ symbol correspond to values left and right of $\mathbf{x}=0$. This boundary condition is implemented in INTERP.

2.4 Continuum Breakdown Surface

As an informative option, the code JET can compute (in PLUMES) points on a surface of continuum breakdown [5,6,7] , which is defined as a line of constant **B**, where **B** is given by :

$$\mathbf{B} = -(\mathbf{u}/\varphi)\, \rho^{-1} (d\rho/dS)$$

$$\varphi = 4(\pi\gamma)^{-1/2}\, \sigma\, n\, a \tag{2.4-1}$$

When the standard isentropic relations for ρ and $\mathbf{n}$ in terms of $\mathbf{M}$ are substituted in (2.4-1), the flow speed is expressed as $\mathbf{u}=\mathbf{Ma}$ and the streamwise gradient of $\mathbf{M}$ is expressed in cartesian coordinates, we get :

$$B = \lambda_0\, (\pi\gamma/8)^{1/2}\, M^2 \left[1 + ((\gamma-1)/2)M^2\right]^{1/(\gamma-1)\,-\,1} \left[M_x \cos\theta + M_y \sin\theta\right]$$

$$\lambda_0 = \left(2^{1/2}\, \sigma\, n_0\right)^{-1} \tag{2.4-2}$$

Note that the sign of **B** has been chosen as positive for expansion flows. This definition is preferred to taking an absolute value of the flow gradient, since it assures proper interpolation of **B** even if its spatial distribution goes through $\mathbf{B}=0$.

Due to the dependence of **B** on a spatial gradient, its numerical evaluation (see BREAK) is attributed to mid-grid points both in $\mathbf{x}$ and in $\mathbf{y}$.

3. THE JET CODE

In this chapter we provide a concise description of the JET code according to its version at the time of the JET018 run. This description is intended as an aid in reading the code listing which is given in Ch. 4.

The plan of this chapter is as follows. Array variables that constitute the mainstay of the computational scheme are described in Section 3.1. Auxiliary array variables that are used primarily for processing the information generated by the numerical scheme, are described in Section 3.2, followed in Section 3.3 by a list of major parameters that control the computation (some of them also serve as run data). Finally, all subroutines are listed and described in Section 3.4.

3.1 Main Variables

The array variables used for the computational scheme are organized in two labeled COMMON groups. The first group /VECS/ is designed to hold two grid rows - the old row designated by suffix F and the new row designated by suffix N. The second group /CHARAC/ are characteristic-indexed arrays that hold information about continuous characteristic lines. This characteristic information is used in two ways : it is incorporated in the SIMA computational scheme for the CRW region, and it is used to store data for optional plotting of characteristic lines (see PLUMES and PRINT).

The basic organization is that the new arrays (suffix N) are those in which values are stored during the course of the marching computational procedure. At the end of each marching step, values are transferred from new arrays to old arrays (suffix F); this is done in MOVE. In the array listing below, we indicate in parenthesis the subroutine (or subroutines) in which that new array is defined.

/VECS/

XN(I)	x coordinate of grid point I. (GRIDN)
RMN(I)	modified Riemann invariant $(\nu+\theta)$ at grid point I. (BEGIN, INVMAR, LOADC).
RPN(I)	modified Riemann invariant $(\nu-\theta)$ at grid point I. (BEGIN, INVMAR, LOADC).
MN(I)	Mach number at grid point I (BEGIN, INVMAR, LOADC).
MUN(I)	Mach angle μ at grid point I. (BEGIN, INVMAR, LOADC).
TETAN(I)	true (unmodified) flow angle θ at grid point I. (BEGIN, INVMAR, LOADC).

BN(I)	value of breakdown parameter **B** at point I-1/2 (and at half a marching step back in **y** as well). (BREAK).
XTEMP(I)	used for auxiliary computation of I-1/2 grid points in PLUMES.

/CHARAC/

XCHARN(KC)	**x** coordinate of point on characteristic line number KC. (BEGIN, SEMINV, PLUMES).
YCHARN(KC)	**y** coordinate of point on characteristic line number KC. (BEGIN, SEMINV, PLUMES).
RMCARN(KC)	modified Riemann invariant $(\nu+\theta)$ of point on characteristic line number KC. (BEGIN, SEMINV).
RPCARN(KC)	modified Riemann invariant $(\nu-\theta)$ of point on characteristic line number KC. (BEGIN, SEMINV).
TCHARN(KC)	true (unmodified) flow angle θ at point on characteristic line number KC. (BEGIN, SEMINV).
MUCARN(KC)	Mach angle μ at point on characteristic line number KC. (BEGIN, SEMINV).
CSIGNN(KC)	sign of characteristic line number KC. It has value 1 for C^+ and value -1 for C^-. Note that upon reflection of a C^+ line from the symmetry plane $(x=0)$, the sign value is changed from 1 to -1. (BEGIN, SEMINV).
MCHARN(KC)	Mach number at point on characteristic line number KC. (BEGIN, SEMINV).
MCHARI(KC)	Mach number at Prandtl-Meyer's fan characteristic number KC at the corner. It is defined initially and is not changed during the run. (BEGIN).

3.2 Auxiliary Variables

In addition to the major arrays mentioned above, there are several groups of auxiliary arrays that do not affect the computational scheme, but are intended for informative processing of the results. These groups are /PLUME/, /IPLUME/, /THICKY/, /THICKX/, /GRP/. /PLUME/ is used to preserve points on special lines for later plotting (in a separate code). /THICKY/ and /THICKX/ are for storing values of radial (**y**) and lateral (**x**) molecular opacities. The group /GRP/ is used in conjunction with comparative computation of the ring-symmetric CRW flow according to the analytic approximation [1] .

/PLUME/ (PLUMES, PRINT)

XPL(J,IPL)	**x** coordinate at marching step J of special line number IPL.
YPL(J,IPL)	**y** coordinate at marching step J of special line number IPL.

/IPLUME/ (PLUMES, PRINT)

KPL	number of special lines computed in PLUMES.
ITYPL(IPL)	index indicating the type of special line number IPL.

/THICKY/ (OPACY, PRINT)

XTH(J)	**x** coordinate on boundary characteristic line at marching step J, from which radial opacity is integrated.
TH(J)	radial opacity computed by **y**-integration from the boundary point defined by XTH(J) (up to current YN).

/THICKX/ (OPACX, PLUMES, PRINT)

YXI(JXI)	**y** coordinate of printed row number JXI (the index JXI counts just rows that have been printed). The row to be printed next upon calling PRINT is the row having YF near YXI(JXI).
XI(I,JXI)	lateral (**x**) molecular opacity [1] at point XF(I), for printed row JXI. It is obtained by numerically integrating the solution obtained from the JET computation (see OPACX).
XIPM(I,JXI)	same as XI(I,JXI) except that the Prandtl-Meyer solution is used to estimate the flow at grid points XF(I).
XIGRP(I,JXI)	same as above, except that the analytic approximation to a ring-symmetric CRW [1] is used to estimate the flow at grid points XF(I).
XIAPP(I,JXI)	same as XIGRP(I,JXI) except that the numerical integration is replaced by an approximate closed-form expression [1] .
XIF(I,JXI)	stores grid points XF(I) of printed row JXI.

/GRP/ (PRINT, HMSET, MFUNC, HINTER, MATCH)

DMINV	increment of inverse Mach number for array MHINV(I).
MHINV(I)	inverse Mach number array (from 0 to 1/MEXIT), from which the **H**(M) function can be evaluated (HMSET).
HMV(I)	values of the **H(M)** function evaluated by numerical integration. It is used to compute this function by interpolation. (HMSET, HINTER).

3.3 Major Parameters

Parameters that define and control a particular run (such as the maximum **y** for the marching computation, the number of grid points on a row and many more) are defined in INIDAT. (The code JET has no input file and no READ statements). The major control parameters are grouped in /PAR/ (floating point) and in /IPAR/ (integers); thermodynamic data are grouped in /STAG/.

We indicate in the listing the subroutines in which the labeled COMMON group or a particular parameter is defined (or sometimes referred to).

/PAR/ (INIDAT)

MEXIT	nozzle exit Mach number ($\mathbf{M}_l$).
MFIN	Mach number of the final (boundary) CRW characteristic at the corner ($\mathbf{M}_f$).
YMAX	maximum value of **y** for the marching scheme. When YF.GE.YMAX the run is terminated.
DY0	initial marching step.
DY	current marching step.
DYNEXT	next marching step (YSTEP).
STAB	stability coefficient for marching step (STAB.LE.1). (See YSTEP).
DELTA	symmetry index. DELTA = 0 for plane symmetry; DELTA = 1 for ring-symmetry.
PSI1	angle of Prandtl-Meyer fan characteristic at exit conditions (measured from **x** axis).
PSIF	angle of final (boundary) Prandtl-Meyer fan characteristic.
SIGMA	collision cross-section (σ).
FRACG	the number of intervals initially allocated to the CRW fan is a FRACG fraction of the total number of intervals (KF0-1). (see BEGIN).
EPSIL	convergence parameter (small number). (INVMAR, SEMINV).
TETLIM	flow angle (from **x** axis) of the limiting ($\mathbf{p} = \mathbf{0}$) velocity vector of the flow at the lip-centered Prandtl-Meyer fan.
TETSYM	PAI-2*TETLIM for reflection transformation (see INTERP).

/IPAR/ (INIDAT)

JMAX	maximum number of marching steps. If J.GE.JMAX run is terminated.
KF0	initial (and maximum) number of grid points in a row.
KF	current number of grid points in the old row.
KN	current number of grid points in the new row.

ITER0	maximum number of iterations for the integration of the compatibility relations (see INVMAR and SEMINV; also used in RFUNC, PLUMES).
IM, IP	search indices for interpolation subroutine INTERP. (see INVMAR, SEMINV).
J	current row index (also index of a marching step).
KF2	defined as 2*KF; not used in present version.
IDEL, JDEL	increments for printing grid point I and row J (see PRINT).
JYXI	number of rows to be printed in a run.
JXI	index of printing row, to be printed next (see PRINT).
ILEAD	index I at the first grid point on current new row, where the SIMA integration commences. Initially this point corresponds to the leading characteristic of the CRW. (see GRIDN, BEGIN).
ILEADF	value of ILEAD for current old row.
KCLEAD	index in the characteristic array for the characteristic line that corresponds to the new grid point I = ILEAD (see GRIDN). Initially KCLEAD = 1.

/STAG/ (INIDAT)

RHO0, N0	stagnation density and number density.
P0, T0, A0	stagnation pressure, temperature and sound speed.
MDOT1	mass flow rate from ring-nozzle (only from the $x > 0$ half). (See PRINT).

/ICHARA/ (BEGIN)

KCHARP	number of C^+ characteristic lines for which data is stored (either for SIMA computation or for subsequent plotting).
KCHARM	number of C^- characteristic lines for which data is stored (only for subsequent plotting).
KCHAR0	total number of characteristics for which data is stored, i.e., KCHAR0 = KCHARP + KCHARM.

3.4 Description of JET subroutines

MAIN PROGRAM

The main program performs two functions. The first section (up to statement 1) is the initial set up; it is performed just once. The second section is the marching loop with the step index J. This program can be read as a flow chart of the overall computational procedure.

INIDAT is for setting up run data. In BEGIN the initial conditions for the marching computation are set up. A single marching step is performed by calling MARCH, and the loading of new row vectors into old row vectors is done by calling MOVE. The call to YSTEP is for the first computed marching step. All remaining calls are for informative tasks (see HMSET, BREAK, OPACY, PLUMES, PRINT). Run is terminated when either YF.GE.YMAX or when J.GE.JMAX.

NOTE ON EXEC : The only special feature in the EXEC is retaining the output unit 7 file for optional post-plotting. The printed output (unit 6) is the system's standard (default).

INIDAT

Initial data definition and preliminary data computations. The data is defined by statements rather than by reading an input file. The meaning of major parameters was described in Section 3.3 above. User is invited to modify the data definitions, particularly of run-control parameters such as YMAX, JYXI and YXI(JXI) (for printing JYXI selected rows).

BEGIN

Here all initial values (prior to beginning of marching schemes integration) are loaded into all major computational arrays (Section 3.1). Also, values of the key integer parameters KCHARP, KCHARM, KCHAR0, ILEAD, KCLEAD and KF are defined.

In the first loop (loop 1) we define an initial family of $\mathbf{C}^+$ characteristic lines for the lip-centered CRW, by storing the Mach number of the Prandtl-Meyer fan characteristics in the array

MCHARI(KC). Note that the fan characteristics are generated at equal RP intervals, since the flow variables are RM and RP. However, a different division might also be acceptable.

The next step is the definition of initial values for all characteristic arrays, first the C^+ arrays (loop 2), then the C^- arrays (loop 21). The C^- characteristic lines are needed just for informative output (post-plotting), so the present version contains just one C^- line. The user may modify that.

The remaining grid points (altogether KF0 grid points are initially available) are uniformly distributed across the nozzle opening, and the row arrays are loaded with the corresponding nozzle-exit flow variables (loop 3).

PRINT

The main task of this subroutine is the printing of flow variables at grid points of selected rows. The printing of a row is selected when YF is close to a predefined array YXI(JXI). Following the printing, JXI is updated by adding 1.

For comparison, additional flow variables are printed for each row. These are computed from the analytic approximation to a ring-symmetric CRW [1] , by calling MATCH. Also, lateral molecular opacities of various kinds of approximation are computed by calling OPACX, and are printed for each grid point within the CRW.

Following the row printing (statement 120), arrays intended for post-processing (plotting of special lines) are printed and subsequently written on output unit 7. This is done once per run, just before run termination.

FIN

This subroutine is called when an error is encountered, in order to terminate the run. Note that the run is terminated by deliberately introducing an error of computing SQRT(-1), which is done in order to trigger the printing of calling sequences by the operating system.

MARCH

This subroutine performs a single marching step by calling the proper computational subroutines at an appropriate sequence. It can be read as a flow chart of the entire computational scheme. First the segment of the new row suitable for SIMA computation is calculated by calling SEMINV. Then new grid points for that part of the new row for which inverse marching integration is to be performed, are generated by calling GRIDN. The results of the SEMINV computation, which were stored in characteristic arrays, are now loaded into row arrays by calling LOADC. Finally, the computation of the new row is completed by calling INVMAR which computes the flow at the remaining grid points by the inverse marching scheme.

INVMAR

This is one of the two central subroutines for computing the flow at new grid points (the other is SEMINV). Here the inverse marching scheme is used. The computational procedure follows the seven-step description given in Section 2.2 above. Note that the initial value of the search indices IM and IP is not redefined at each call to INTERP, since it is assumed that IM and IP do not change much at consecutive calls to INTERP, so that search efficiency is enhanced by not starting the search from an arbitrary point (such as either end of the row).

SEMINV

This is the subroutine performing the SIMA scheme for computing the flow at new grid points located along continuous characteristic lines of the lip-centered CRW (at prescribed $\dot{y}$-marching steps). The essence of the computational procedure of this subroutine was given as a seven-step description in Section 2.2. The same remark about IM given in the preceding INVMAR description applies here as well.

The main loop (100) is over all characteristic lines, including some C^- lines in addition to the C^+ lines. Thus, the array CSIGNF(KC) is used to get the appropriate expressions for either C^+ or C^- characteristics. It is noted that while normally the characteristic segments through points 1 and 2 are C^- and C^+ respectively, this is reversed when a C^- rather than a C^+ line is computed via the SIMA scheme. In this case, which is characterized by having CSIGNF(KC).LT.0, the Riemann

invariants integrated along segments (1,4) and (2,4) are interchanged. This is done in the few statements just preceding and following statement 21.

An additional capability of this subroutine is to treat a change of a C^+ characteristic line into a C^- line upon reflection from the symmetry plane ($x = 0$). This is done by first computing a new grid point having X4.LT.0, and then changing its sign after setting CSIGNN(KC) = −1 (statements just preceding statement 30). It is also possible to skip the computation of a particular characteristic by setting CSIGNN(KC) = 0. This feature is not exploited in the present version.

Finally, we note that not all characteristic lines computed here are part of the marching flow computation. Only those with indices KC between KCLEAD and KCHARP are. All other characteristic lines are computed just for informative purposes (post-plotting).

RFUNC

Here M, MU, TETA are computed from the two Riemann invariants RM, RP. The computation of M is performed by a Newton-Raphson iteration using Equations (2.1-2) and (2.1-3) given in Section 2.1 above.

INTERP

This subroutine starts by finding through a search procedure the grid interval (I, I+1) that contains a given point X. Then the Riemann invariants are computed for this point by linear interpolation, and returned in RM, RP. Note that X may be negative, which accounts for the relatively elaborate search logic in the determination of I, and for the reflection transformation (as in Eq. (2.3-1) above) preceding the last two statements of the subroutine.

INTERX

This interpolation routine performs an inverse task to that of INTERP, in that it finds the point X0 that corresponds to a given linearly interpolated value of the flow variable VAR0. It is used in PLUMES to compute the location of a breakdown surface point on a new row of **x**-centered and **y**-centered grid points

BREAK

This subroutine computes the new breakdown parameter array BN(I). The computation is based on the description given in Section 2.4 above.

OPACY

Here the radial (Y) molecular opacity array TH(J) is computed. At each marching step J, a new boundary grid point XTH(J) is added, then the radial opacities at all preceding boundary points are updated by adding the contribution of the gas layer between the current old and new rows. Note that since grid points on adjacent rows are not located on equal-**x** columns, this procedure requires **x**-interpolation by calling INTERP.

PLUMES

This is a user-defined subroutine, where up to 10 special lines can be computed and subsequently retained on output unit 7 for post-processing (plotting). The type of the line ITYPL(IPL) and a parameter VPL(IPL) that defines the line, are computed through user-inserted statements in the section preceding statement 2000. Then an additional point on the current new row is computed for each line type. The available types are clearly stated in comments. Note that characteristic lines have already been computed in SEMINV using the SIMA scheme, regardless of whether they are part of the solution grid to the flow field, or are just computed for informative purpose. It is the user's choice which of these lines (if any) are to be saved in the /PLUME/ arrays for subsequent post-processing (plotting).

GRIDN

This subroutine computes the grid points in that segment of the new row for which the flow is computed by the inverse marching scheme (in INVMAR). Initially, this segment extends from $\mathbf{x}=0$ to the new row grid point which lies on the leading characteristic of the lip-centered CRW. However, since the leading characteristic is reflected from the symmetry plane ($\mathbf{x}=0$) at some point, this segment steadily shrinks in size as the marching proceeds. The remedy is to declare the next-to-the-

leading characteristic line (KC = 2) as the beginning of the segment for SIMA integration, by setting KCLEAD = 2. This process of increasing KCLEAD is repeated whenever it is deemed necessary. The criterion in the present version for the minimal KCLEAD is that the inverse-marching segment should be at least twice DX1 - the average CRW grid interval (loop 1, the two statements following DX1 = ...). Also, ILEAD is redefined for each row according to XLEAD/DX1 + 2 in order to achieve a row of relatively uniform grid intervals throughout. The result is that the number of grid points in a row is initially KF0, but eventually it decreases due to both increase of KCLEAD and decrease of ILEAD.

YSTEP

In this subroutine the next marching step DYNEXT is computed at the end of the current marching step. It is defined as the smallest step obtained by forward intersection of C^- and C^+ characteristics from adjacent grid points. Note that the actual value of DYNEXT is reduced by a "stability" factor STAB, and that $\dot{D}Y$ is also limited by the growth-rate factor DDY and by DYMAX (see MAIN PROGRAM).

MOVE

Here old row arrays (loop 1) and old characteristic arrays (loop 2) are loaded with values of flow variables from corresponding new arrays, in preparation for the next marching step. As a result of this organizational feature, informative computations (e.g. BREAK, OPACY) that require both new and old rows, have to be performed prior to calling MOVE.

OPACX

Here lateral (X) opacities that correspond to the number of expected collisions of a fast molecule invading the CRW in the $-X$ direction, are computed. All opacities, except XIAPP(1), are computed by numerical integration. In loop 1 we compute the opacity contribution of the segment lying just outside the computational boundary characteristic (MFIN), assuming a Prandtl-Meyer flow. This additional opacity is denoted XI0. If the flow is ring-symmetric, XI0 is recalculated using the analytic approximation [1] to estimate the flow field at the fringes of the ring-symmetric CRW (see also the closed form expression for τ in [1]).

The computation of opacity arrays starts after statement 14. First, the opacity at each grid point is set to XI0. Thus, even though the numerical flow computation does not include the fluid outside the boundary characteristic line, the opacity integration includes an estimate of that "missing" part, i.e., of XI0. In typical case computations of a ring-symmetric CRW [1] we found that the maximum value of XI0 was about 0.16., which indicated that as far as interaction with invading ambient molecules is concerned, the approximation MFIN = 34 was a reasonable substitute for MFIN = ∞.

The next step is the computation by numerical integration of three approximations to the lateral opacity : XI(I,JXI), XIPM(I,JXI), XIGRP(I,JXI). (Note that when the flow is ring-symmetric, the approximation XIPM(I,JXI) obtained by assuming a Prandtl-Meyer flow is usually grossly exaggerated). The opacity XIGRP(I,JXI) is based on the analytic approximation to a ring-symmetric CRW [1] , and is reasonably close to XI(I,JXI) which is obtained from the numerical solution to the flow field. Finally, a simplified closed-form integration of lateral opacity [1] is computed as XIAPP(I,JXI) (loop 3). Thus, the quantitative difference between XI(I,JXI) and XIGRP(I,JXI) is an indication to the degree of accuracy achieved by the analytic approximation to a ring-symmetric CRW [1] , while the difference between XIGRP(I,JXI) and XIAPP(I,JXI) indicates the level of error introduced by the closed-form integration of lateral opacity [1] .

LOADC

Here flow variables of new grid points computed via the SIMA scheme (SEMINV) are loaded into new row arrays from corresponding characteristic arrays.

NUFUNC

This function computes the modified $\nu(M)$ value as given by Eq. (2.1-2). Note that presently NU0 = 0 (see INIDAT).

HMSET

This subroutine is called just once from the MAIN PROGRAM. Its task is to set up the arrays in /GRP/, so that the function H(M) [1] can be evaluated by interpolation (in HINTER). There is also an informative printout of various derivatives (see Ch. 3 of [1]) generated in this subroutine.

MFUNC

This subroutine is called by HMSET in order to compute functions of Mach number that serve in the computation of **H(M)**. The output variable F is the integrand for the integration leading to **H(M)**.

HINTER

This subroutine computes **H(M)** by linear interpolation in inverse Mach number, using the /GRP/ arrays computed in HMSET.

MATCH

This subroutine is called from PRINT to compute the Mach number according to the analytic approximation of a ring-symmetric CRW [1] , for point (YF,XF(I)). M0B is the associate Mach number $\mathbf{M(0,\beta)}$, which is preserved in the array MCHARI(KC) for all CRW characteristics that are used in the SIMA computation. Hence the Mach number $\mathbf{M(\alpha,\beta)}$, denoted by MAB can be computed directly from the analytic approximation [1] to the area function at (YF,XF(I)) by calling AREAF. Since typically $\mathbf{M(0,\beta)}$ is not known, we also compute the Mach number via the inverse-problem procedure [1] , denoting the resulting Mach numbers by suffix I : M0BI for $\mathbf{M(0,\beta)}$ and MABI for $\mathbf{M(\alpha,\beta)}$. The inverse-problem iterative procedure [1] is performed in loop 1, resulting in M0BI. From M0BI the value of MABI is computed through the area function approximation as for MAB above.

AREAF

This subroutine computes the Mach number M that corresponds to the area function F (Eq. (3.2-1) of [1]). The computation is done by Newton-Raphson iterations, and it has been found to converge when M.GT.1 (and when $M - 1$ is not much smaller than 1).

4. THE JET CODE LISTING

```
C$OPTIONS LIST
C JET018
C   "JET"  A SEMI-INVERSE MARCHING CHARACTERISTICS METHOD FOR RING JETS.
C   USING RIEMANN INVARIANTS RM=(NU+TETA), RP=(NU-TETA) AS FIELD
C   VARIABLES.
      IMPLICIT REAL*8(A-H,L-Z,$)
      REAL*4 XPL,YPL
      COMMON /PLUME/XPL(1002,10),YPL(1002)
      COMMON /IPLUME/KPL,ITYPL(10)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1             TETAF(101),BF(101),
     2             XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3             TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      REAL*4 YXI,XI,XIPM,XIGRP,XIAPP,XIF
      COMMON /THICKX/YXI(20),XI(101,20),XIPM(101,20),XIGRP(101,20)
     1              ,XIAPP(101,20),XIF(101,20)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1           STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2           TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1               RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2               TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3               CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4               MCHARI(92)
      COMMON /ICHARA/KCHARP,KCHARM,KCHAR0
      COMMON /GRP/DMINV,MHINV(101),HMV(101)
      COMMON /IGRP/KHM
C
      PRINT 101
  101 FORMAT('1')
      J=1
      IF(J.EQ.1) STOP
      CALL INIDAT
      PRINT 101
      CALL HMSET
      PRINT 101
      CALL BEGIN
      CALL MARCH
      CALL OPACY
      CALL PLUMES
      CALL PRINT
      J=2
      CALL PLUMES
      CALL MOVE
      CALL OPACY
      CALL PRINT
      CALL YSTEP
 1    J=J+1
C   DY WAS DETERMINED BY THE PREVIOUS CALL TO GRIDN.
      DY=DMIN1(DYNEXT,DY*DDY,DYMAX)
C   INTEGRATE BY ONE Y-STEP
      CALL MARCH
C   BREAKDOWN PARAMETER (BF(I)).
      CALL BREAK
C   SPECIALLY DESIGNATED LINES (FOR PLOTTING).
      CALL PLUMES
C   STORE NEW LINE (N) IN OLD LINE (F).
      CALL MOVE
C   COMPUTE RADIAL MOLECULAR OPACITIES
      CALL OPACY
C   Y-STEP IS VARIABLE, SO JMAX IS USED AS END-OF-RUN CRITERION.
      IF(YF.GE.YMAX) JMAX=J
C   PRINT FIELD AT MOST RECENT Y.
      CALL PRINT
C   NEXT Y-STEP.
```

```
      CALL YSTEP
      IF(J.LT.JMAX) GO TO 1
      STOP
      END
```

INIDAT

```
      SUBROUTINE INIDAT
C  SUBROUTINE NUMBER  1
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1            TETAF(101),BF(101),
     2            XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3            TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      REAL*4 YXI,XI,XIPM,XIGRP,XIAPP,XIF
      COMMON /THICKX/YXI(20),XI(101,20),XIPM(101,20),XIGRP(101,20)
     1              ,XIAPP(101,20),XIF(101,20)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1            G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1           STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2           TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1            KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
C
      PAI=4.D0*DATAN(1.D0)
      PAI2=2.D0*DATAN(1.D0)
      DEG=180.D0/PAI
      AR=8.3143D3
      AV=6.022D 26
      AW=7.27D0
      RHO0=0.0075D0
      T0=2300.D0
      G=1.54D0
      D=2.5D-10
      MEXIT=4.D0
      MFIN=34.D0
      XC=0.5D0
      YC=2.5D0
C  DELTA=0 CORRESPONDS TO PLANE SYMMETRY
C  DELTA=1 CORRESPONDS TO CYLINDRICAL SYMMETRY
      DELTA=1.D0
      FRACG=0.6D0
      EPSIL=1.D-8
      ITER0=20
      KF0=101
      JMAX=1001
      STAB=0.50D0
      DDY=1.05D0
      DYMAX=0.5D0
      YMAX=50.D0
      DY0=YC/250.D0
      IDEL=1
      JDEL=1
C  POINTS FOR PRINTING FLOW FIELD AT YF=YXI(JXI)
      JXI=1
      JYXI=11
      DYXI=5.D0
      YXI(1)=YC+0.5D0
      YXI(2)=YXI(1)+2.D0
      I0=2
      DO 1 I=I0,JYXI
      YXI(I)=YXI(I0)+DYXI*DFLOAT(I-I0)
 1    CONTINUE
      IF(KF0.GT.101) CALL FIN(101)
      IF(JMAX.GT.1001) CALL FIN(102)
      IF(FRACG.GT.1.D0 .OR. FRACG.LT.0.) CALL FIN(103)
      IF(JYXI.GT.20) CALL FIN(104)
      IF(DELTA*(1.D0-DELTA).NE.0.) CALL FIN(105)
      N0=RHO0*AV/AW
      A0=DSQRT(G*AR*T0/AW)
      P0=AR*RHO0*T0/AW
```

```
      SIGMA=PAI*D**2
      LAMDA0=1.D0/(DSQRT(2.D0)*SIGMA*N0)
      G1=(G-1.D0)/2.D0
      G2=(G+1.D0)/(2.D0*(G-1.D0))
      G3=G/2.D0
      G4=(G+1.D0)/(G-1.D0)
      G5=DSQRT((G+1.D0)/(G-1.D0))
      G6=1.D0/(G-1.D0)
      G7=2.D0/(G+1.D0)
      G8=(0.5D0*(G+1.D0)**2/(G-1.D0))**(1.D0/(G+1.D0))*
     1   ((G+1.D0)/(G-1.D0))**((G-1.D0)/(G+1.D0))
      G9=(G+3.D0)/(2.D0*(G-1.D0))
      G10=(7.D0-3.D0*G)/(2.D0*(G-1.D0))
      G11=(2.D0/(G+1.D0))**(1.D0/(G-1.D0))
      G12=DSQRT((G+1.D0)/(G-1.D0))-1.D0
      G13=(2.D0-G)/(2.D0*(G-1.D0))
      G14=G/(2.D0*(G-1.D0))
      G15=(G+1.D0)/(3.D0-G)
      G16=(G+1.D0)/4.D0
      G20=LAMDA0*DSQRT(PAI*G/8.D0)
      ZETA1=G5*DATAN(DSQRT(MEXIT**2-1.D0)/G5)
      AMU1=DARSIN(1.D0/MEXIT)
      PSI1=PAI2+AMU1
      ZETAF=G5*DATAN(DSQRT(MFIN**2-1.D0)/G5)
      PSIF=PSI1+ZETA1-ZETAF
      NU0=0.
      TETLIM=NUFUNC(MEXIT)+PAI2-NU0
      PSILIM=TETLIM
      TETSYM=PAI-2.D0*TETLIM
      GOREM=1.D0+G1*MEXIT**2
      RHO1=RHO0/GOREM**G6
      V1=MEXIT*A0/DSQRT(GOREM)
      P1=P0/GOREM**(G/(G-1.D0))
      T1=T0/DSQRT(GOREM)
      YYC=2.D0*PAI*YC
      IF(DELTA.EQ.0.) YYC=1.D0
      MDOT1=YYC*XC*RHO1*V1
C
      PRINT 21,AR,AV,AW,G,RHO0,N0,P0,T0,A0,D
 21   FORMAT(/1X,'THERMODYNAMIC DATA:'/
     1         1X,'AR,AV,AW,G=',2X,2D14.5,2F9.3/
     2         1X,'RHO0,N0,P0,T0,A0,D=',6D13.5)
      PRINT 22,XC,YC,MEXIT,RHO1,P1,T1,V1,MDOT1,PSI1*DEG,PSIF*DEG,
     1          PSILIM*DEG
 22   FORMAT(/1X,'CORNER DATA:  XC,YC=',2F9.2/
     1         1X,'EXIT CONDITIONS:',
     2         2X,'MEXIT,RHO1,P1,T1,V1,MDOT1=',F9.3,5D13.4/
     3         1X,'CENTERED FAN LIMITS:',
     4         2X,'PSI1,PSIF,PSILIM=',3F10.3)
      PRINT 23,DELTA,KF0,JMAX,ITER0,DY0,YMAX,STAB,DDY
 23   FORMAT(/1X,'INTEGRATION DATA.      SYMMETRY INDEX:  DELTA=',F4.1/
     1         1X,'NUMBER OF POINTS IN X AND Y DIRECTIONS:  KF0,JMAX=',
     2           2I5/
     3         1X,'MAX. NUM. OF ITERATIONS   ITER0=',I5/
     5         1X,'INITIAL Y-STEP AND MAXIMUM Y:  DY0,YMAX=',2D14.5/
     6         1X,'Y-STEP STABILITY FACTORS  STAB,DDY=',2F7.3)
      RETURN
      END                                             BEGIN
      SUBROUTINE BEGIN
C  SUBROUTINE NUMBER  2
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1            TETAF(101),BF(101),
     2            XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3            TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1            G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1           STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2           TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
```

```
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1               RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2               TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3               CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4               MCHARI(92)
      COMMON /ICHARA/KCHARP,KCHARM,KCHAR0
C
C  DEFINE INITIAL CHARACTERISTIC PARAMETERS.  USE INTERPOLATION OF
C  RIEMANN INVARIANT ACROSS THE FAN.
      KCHARP=IDINT(FRACG*DFLOAT(KF0-1)+1.D-6)+1
      KCHAR0=KCHARP+1
      KCHARM=KCHAR0-KCHARP
      IF(KCHARP.LT.2 ) CALL FIN(200)
      IF(KCHAR0.GT.92) CALL FIN(210)
      IF(KCHARM.LT. 1) CALL FIN(205)
      NU1=NUFUNC(MEXIT)
      RM1=NU0
      TET1=RM1-NU1
      RP0=NU1-TET1
      NUFIN=NUFUNC(MFIN)
      RPFIN=NUFIN-(RM1-NUFIN)
      DRP=(RPFIN-RP0)/DFLOAT(KCHARP-1)
      DO 1 KC=1,KCHARP
      RP1=RP0+DRP*DFLOAT(KC-1)
      CALL RFUNC(RM1,RP1,M1,MU1,TETA1)
      MCHARI(KC)=M1
 1    CONTINUE
C  DATA FOR C+ CHARACTERISTICS.
C  THE RIEMANN INVARIANTS ARE DEFINED IN SUCH A WAY THAT BOTH VANISH AT
C  INFINITE MACH NUMBER.
      RM1=NU0
      DO 2 KC=1,KCHARP
      CSIGNF(KC)=1.D0
      XCHARF(KC)=XC
      YCHARF(KC)=YC
      IF(MCHARI(KC).EQ.0.) CALL FIN(231)
      NU=NUFUNC(MCHARI(KC))
      TET=RM1-NU
      RP1=NU-TET
      CALL RFUNC(RM1,RP1,M1,MU1,TETA1)
      MCHARF(KC)=M1
      MUCARF(KC)=MU1
      TCHARF(KC)=TETA1
      RMCARF(KC)=RM1
      RPCARF(KC)=RP1
 2    CONTINUE
C  DATA FOR C- CHARACTERISTICS.
      KC1=KCHARP+1
      XCHARF(KC1)=0.8D0*XC
      DO 21 KC=KC1,KCHAR0
      CSIGNF(KC)=-1.D0
      MCHARI(KC)=MEXIT
      MUCARF(KC)=DARSIN(1.D0/MCHARI(KC))
      TCHARF(KC)=PAI2
      YCHARF(KC)=YC
      MCHARF(KC)=MEXIT
      RMCARF(KC)=RM1
      RPCARF(KC)=NUFUNC(MEXIT)-(TCHARF(KC)-TETLIM)
 21   CONTINUE
C  DEFINE GRID AND INITIAL CONDITIONS AT EXIT PLANE.
      KFAN=KCHARP-1
      ILEAD=KF0-KFAN
      KCLEAD=1
      KF=KF0
      KF2=2*KF
      YF=YC
      DO 3 I=1,KF
      KC=KCLEAD+I-ILEAD
      IF(KC.GT.KCHARP) CALL FIN(241)
```

```
      IF(KC.GE.1) GO TO 31
      XF(I)=DFLOAT(I-1)*XC/DFLOAT(ILEAD-1)
      MF(I)=MEXIT
      TETAF(I)=PAI2
      GO TO 32
 31   CONTINUE
      XF(I)=XC
      MF(I)=MCHARF(KC)
      TETAF(I)=TCHARF(KC)
 32   CONTINUE
      RMF(I)=NUFUNC(MF(I))+(TETAF(I)-TETLIM)
      RPF(I)=NUFUNC(MF(I))-(TETAF(I)-TETLIM)
      MUF(I)=DARSIN(1.D0/MF(I))
      BF(I)=0.
 3    CONTINUE
      DY=DY0
      DO 4 KC=1,KCHAR0
      CSIGNN(KC)=CSIGNF(KC)
 4    CONTINUE
      DO 5 I=1,KN
      BN(I)=0.
 5    CONTINUE
      RETURN
      END
```

PRINT

```
      SUBROUTINE PRINT
C  SUBROUTINE NUMBER  3
      IMPLICIT REAL*8(A-H,L-Z,$)
      REAL*4 XPL,YPL
      COMMON /PLUME/XPL(1002,10),YPL(1002)
      COMMON /IPLUME/KPL,ITYPL(10)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1             TETAF(101),BF(101),
     2             XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3             TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      REAL*4 YXI,XI,XIPM,XIGRP,XIAPP,XIF
      COMMON /THICKX/YXI(20),XI(101,20),XIPM(101,20),XIGRP(101,20)
     1               ,XIAPP(101,20),XIF(101,20)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1               RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2               TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3               CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4               MCHARI(92)
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
C
      SUM=0.
      KF1=KF-1
      DO 10 I=1,KF1
      DX=XF(I+1)-XF(I)
      GOREM=1.D0+G1*MF(I)**2
      GOREM1=1.D0+G1*MF(I+1)**2
      RATEM=RHO0*A0*MF(I  )*DSIN(TETAF(I  ))/GOREM **(G6+0.5D0)
      RATEP=RHO0*A0*MF(I+1)*DSIN(TETAF(I+1))/GOREM1**(G6+0.5D0)
      SUM=SUM+DX*(RATEM+RATEP)/2.D0
 10   CONTINUE
      YYF=2.D0*PAI*YF
      IF(DELTA.EQ.0.) YYF=1.D0
      MDOTFR=YYF*SUM/MDOT1
      PRINT 11, J,KCLEAD,KF,ILEAD,YF,DY,XF(KF),MF(KF),MDOTFR
 11   FORMAT(1X,'J,KCLEAD,KF,ILEAD,YF,DY,XF(KF),MBOUND,MDOTR=',
     1          4I5,5D12.4)
C
C  PRINT FLOW FIELD AT Y=YF
C
```

```
      IF(J.EQ.JMAX) JXI=MIN0(JXI,JYXI)
      IF(J.EQ.1 .OR. J.EQ.JMAX) GO TO 121
      IF(JXI.GT.JYXI) GO TO 120
      IF(YXI(JXI).GT.YF+0.5D0*DY) GO TO 120
  121 CONTINUE
      YXI(JXI)=YF
      CALL OPACX
C  COMPUTE MACH NUMBER FOR CYLINDRICAL EXPANSION MCYL.
      F=(YF/YC)*(G7*(1.D0+G1*MEXIT**2))**G2/MEXIT
      CALL AREAF(F,MCYL)
      PRINT 22,JXI,KCLEAD,ILEAD,KF,MCYL,YF
   22 FORMAT(/1X,'PRINTING NUMBER  JXI,KCLEAD,ILEAD,KF=',4I4,
     1        5X,'MCYL,YF=',2D14.5/)
      PRINT 1
    1 FORMAT(/1X,'  I ',' KC ','     XF(I)   ','  TETAF(I)  ',
     1                        '     MF(I)   ','    MAB     ',
     2                        '     MABI    ','    MOBI    ',
     3                        '    XI(I)    ','   XIGRP(I) ',
     4                        '   XIAPP(I) ','   XIPM(I)  '/)
      IDEL1=IDEL
      IF(J.EQ.1.OR.J.EQ.JMAX) IDEL1=1
      DO 20 I=1,KF,IDEL1
      KC=KCLEAD+(I-ILEAD)
      IF(KC.LT.KCLEAD) KC=0
      MOB=1.D10
      MOBI=1.D10
      MAB=1.D10
      MABI=1.D10
      MPM=MF(I)
      IF(KC.EQ.0) GO TO 23
      MOB=MCHARI(KC)
      IF(J.EQ.1) GO TO 23
      PSIPM=PAI2-DATAN((XF(I)-XC)/(YF-YC))
      ZETA=PSI1+ZETA1-PSIPM
      MPM=DSQRT((G5*DTAN(ZETA/G5))**2+1.D0)
      CALL MATCH(I,MOB,MAB,MOBI,MABI)
   23 CONTINUE
      PRINT 21,I,KC,XF(I),TETAF(I)*DEG,MF(I),MAB,MABI,MOBI,
     1         XI(I,JXI),XIGRP(I,JXI),XIAPP(I,JXI),XIPM(I,JXI)
   21 FORMAT(1X,2I4,10D12.4)
   20 CONTINUE
      IF(J.EQ.1) GO TO 120
      IF(J.EQ.JMAX) GO TO 120
      JXI=JXI+1
  120 CONTINUE
      IF(J.LT.JMAX) GO TO 200
      PRINT 101
  101 FORMAT('1')
      PRINT 102
  102 FORMAT(1X,'RADIAL MOLECULAR THICKNESS  J,XTH(J),TH(J)='/)
      PRINT 202,(JJ,XTH(JJ),TH(JJ),JJ=1,JMAX)
  202 FORMAT(/5(I5,D11.4,D10.3))
      PRINT 101
      PRINT 103,(IPL,ITYPL(IPL),IPL=1,KPL)
  103 FORMAT(1X,'PLUME TYPES  IPL,ITYPL(IPL)=',
     1          2(/1X,5(5X,2I4)))
      PRINT 104
  104 FORMAT(1X,'PLUME POINTS  J,YPL(J),XPL(J,1),XPL(J,2),...='/)
      JDEL1=1
      DO 203 JJ=1,JMAX,JDEL1
      PRINT 204,JJ,YPL(JJ),(XPL(JJ,IPL),IPL=1,KPL)
  204 FORMAT(1X,I5,2X,E12.4,10E11.3)
  203 CONTINUE
C  WRITE ON TAPE7 FOR SUBSEQUENT PLOTTING.
C  NO MORE THAN 80 CHARACTERS PER LINE ON TAPE7.
      WRITE(7,205) JMAX,KPL
  205 FORMAT(8I10/8I10)
      WRITE(7,205) (ITYPL(IPL),IPL=1,KPL)
      DO 210 JJ=1,JMAX
      WRITE(7,211) YPL(JJ),(XPL(JJ,IPL),IPL=1,KPL)
  211 FORMAT(6E13.6/2X,6E13.6/2X,6E13.6/2X,6E13.6)
  210 CONTINUE
```

```
C  WRITE LATERAL (X) OPACITIES
      JXI0=JXI
      WRITE(7,205) JXI0,KF0
      PRINT 226, JXI0,KF0
  226 FORMAT(///1X,'LATERAL (X) OPACITIES  JXI0,KF0=',2I8)
      DO 220 JXI=1,JXI0
      WRITE(7,221) JXI,YXI(JXI)
  221 FORMAT(I10,E13.6)
      PRINT 227, JXI,YXI(JXI)
  227 FORMAT(//1X,'JXI,YXI(JXI)=',I8,E15.6/)
      DO 225 I=1,KF0
      WRITE(7,211) XIF(I,JXI),XI(I,JXI),XIPM(I,JXI),XIGRP(I,JXI),
     1            XIAPP(I,JXI)
      PRINT 211, XIF(I,JXI),XI(I,JXI),XIPM(I,JXI),XIGRP(I,JXI),
     1          XIAPP(I,JXI)
  225 CONTINUE
  220 CONTINUE
  200 CONTINUE
      RETURN
      END
```

FIN

```
      SUBROUTINE FIN(IFIN)
C  SUBROUTINE NUMBER  4
C  STOP WHEN ERROR IS DETECTED.
      IMPLICIT REAL*8(A-H,L-Z,$)
      PRINT 1,IFIN
    1 FORMAT(/1X,'FIN CODE IFIN=',I6/)
C  INDUCE ERROR IN ORDER TO GENERATE TRACING OF CALLING SUBROUTINES.
      X=-1.D0
      Y=X+DSQRT(X)
      IF(IFIN.LE.0) GO TO 100
      STOP
  100 RETURN
      END
```

MARCH

```
      SUBROUTINE MARCH
C  SUBROUTINE NUMBER  5
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1            TETAF(101),BF(101),
     2            XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3            TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1            G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1           STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2           TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1            KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1              RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2              TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3              CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4              MCHARI(92)
      COMMON /ICHARA/KCHARP,KCHARM,KCHAR0
C
C  ADVANCE FLOW FIELD FROM YF TO YN
      IM=KF
      IP=KF
      YN=YF+DY
      KN=KF0
C  SEMI-INVERSE INTEGRATION FOR FAN POINTS.
      CALL SEMINV
C  NEW GRID POINTS (JUST INVERSE MARCHING).
      CALL GRIDN
C  LOAD FLOW VARIABLES FROM SEMI-INVERSE INTEGRATION INTO VECTORS
      CALL LOADC
C  CHARACTERISTIC SCHEME INTEGRATION FOR INNER POINTS (INVERSE MARCH).
      CALL INVMAR
      RETURN
      END
```

INVMAR

```
      SUBROUTINE INVMAR
C  SUBROUTINE NUMBER  6
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1             TETAF(101),BF(101),
     2             XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3             TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
C
C  INTEGRATION WITH INVERSE CHARACTERISTICS FOR NEW POINT(X4,Y4).
C  OLD POINTS ARE (X1,Y1),(X2,Y2).
C  X1 IS OBTAINED BY INVERSE C- FROM X4
C  X2 IS OBTAINED BY INVERSE C+ FROM X4
C  NOTE THAT X1 MAY BE NEGATIVE (E. G. WHEN X4=0).
      KN1=ILEAD-1
      IF(KN1.LE.0) CALL FIN(601)
      DO 1000 I=1,KN1
      I4=I
      X4=XN(I)
      Y4=YN
      IF4=(IM+IP)/2
      CALL INTERP(0,IF4,KF,X4,XF,RM4,RMF,RP4,RPF)
      CALL RFUNC(RM4,RP4,M4,MU4,TETA4)
      M14=M4
      MU14=MU4
      TETA14=TETA4
      M24=M4
      MU24=MU4
      TETA24=TETA4
      Y1=YF
      Y2=YF
      Y14=(Y1+Y4)/2.D0
      Y24=(Y2+Y4)/2.D0
      X1=1.D10
      X2=1.D10
      RM4=1.D10
      RP4=1.D10
      ITER=0
      GO TO 2
C
C  CORRECTOR
C
 1    ITER=ITER+1
C  AVERAGED PROPERTIES ON C-(14),C+(24) CHARACTERISTICS.
      RM14=(RM1+RM4)/2.D0
      RP14=(RP1+RP4)/2.D0
      RM24=(RM2+RM4)/2.D0
      RP24=(RP2+RP4)/2.D0
C  M14,MU14,TETA14, M24,MU24.TETA24 AVERAGED ON C-,C+ CHARACTERISTICS.
      CALL RFUNC(RM14,RP14,M14,MU14,TETA14)
      CALL RFUNC(RM24,RP24,M24,MU24,TETA24)
 2    CONTINUE
C  NEW X1,X2
      X10=X1
      X20=X2
      X1=X4-DY/DTAN(TETA14-MU14)
      X2=X4-DY/DTAN(TETA24+MU24)
      IF(X2.LT.0.) CALL FIN(670)
      D14=DSQRT((X1-X4)**2+DY**2)
      D24=DSQRT((X2-X4)**2+DY**2)
C  INTERPOLATE OLD DISTRIBUTION FOR RM1,RP1, RM2,RP2 AT X1,X2.
      CALL INTERP(0,IM,KF,X1,XF,RM1,RMF,RP1,RPF)
      CALL INTERP(0,IP,KF,X2,XF,RM2,RMF,RP2,RPF)
```

```
C  NO NEED FOR RE-AVERAGING SINCE IT INTRODUCES ONLY HIGHER ORDER
C  CHANGES INTO THE ITERATION SCHEME.
C  INTEGRATE THE CHARACTERISTIC EQUATIONS FOR RM4,RP4 AT X4,Y4.
      RM40=RM4
      RP40=RP4
      RM4=RM1+DELTA*DSIN(TETA14)*D14/(M14*Y14)
      RP4=RP2+DELTA*DSIN(TETA24)*D24/(M24*Y24)
C  CONVERGENCE TEST
      EPS=(DABS(X1-X10)+DABS(X2-X20))/DY+DABS(RM4-RM40)+DABS(RP4-RP40)
      IF(ITER.GT.ITER0) GO TO 10
      IF(EPS.GT.EPSIL) GO TO 1
      RMN(I)=RM4
      RPN(I)=RP4
      CALL RFUNC(RM4,RP4,MN(I),MUN(I),TETAN(I))
 1000 CONTINUE
      RETURN
 10   CONTINUE
      PRINT 11,I4,KN,IF4,IM,IP,KF,ITER,ITER0,EPS,EPSIL,X1,X2,X4,M14,M24
 11   FORMAT(1X,'SUBR. INVMAR. I4,KN,IF4,IM,IP,KF,ITER,ITER0=',8I5/
     1       1X,'EPS,EPSIL,X1,X2,X4,M14,M24=',7D14.6/)
      CALL FIN(611)
      RETURN
      END                                            SEMINV
      SUBROUTINE SEMINV
C  SUBROUTINE NUMBER  7
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1             TETAF(101),BF(101),
     2             XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3             TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1               RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2               TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3               CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4               MCHARI(92)
      COMMON /ICHARA/KCHARP,KCHARM,KCHAR0
C
C  COMPUTE NEW POINT (X4,Y4), BY PASSING A C+ CHARACTERISTIC
C  THROUGH OLD POINT (X2,Y2).  BOTH POINTS ARE ON CHARACTERISTIC LINE
C  NUMBER KC.
      IM=1
      DO 100 KC=1,KCHAR0
      IF(CSIGNN(KC).EQ.0.) GO TO 100
C
C  PREDICTOR
C
      Y1=YF
      Y2=YF
      Y4=YN
      Y14=(Y1+Y4)/2.D0
      Y24=(Y2+Y4)/2.D0
      X2=XCHARF(KC)
      RM2=RMCARF(KC)
      RP2=RPCARF(KC)
      M2=MCHARF(KC)
      MU2=MUCARF(KC)
      TETA2=TCHARF(KC)
      M14=M2
      MU14=MU2
      TETA14=TETA2
      M24=M2
      MU24=MU2
```

```
      TETA24=TETA2
      X4=1.D10
      X1=1.D10
      RM4=1.D10
      RP4=1.D10
      ITER=0
      GO TO 2
C
C  CORRECTOR
C
 1    ITER=ITER+1
C  AVERAGED PROPERTIES ON C-(14),C+(24) CHARACTERISTICS.
      RM14=(RM1+RM4)/2.D0
      RP14=(RP1+RP4)/2.D0
      RM24=(RM2+RM4)/2.D0
      RP24=(RP2+RP4)/2.D0
C  M14,MU14,TETA14, M24,MU24,TETA24 AVERAGED ON C-,C+ CHARACTERISTICS.
      CALL RFUNC(RM14,RP14,M14,MU14,TETA14)
      CALL RFUNC(RM24,RP24,M24,MU24,TETA24)
  2   CONTINUE
C  NEW X4,X1
      X40=X4
      X10=X1
      X4=X2+DY/DTAN(TETA24+CSIGNF(KC)*MU24)
      X1=X4-DY/DTAN(TETA14-CSIGNF(KC)*MU14)
      D14=DSQRT((X1-X4)**2+DY**2)
      D24=DSQRT((X2-X4)**2+DY**2)
C  INTERPOLATE OLD DISTRIBUTION FOR RM1,RP1, AT X1.
      CALL INTERP(0,IM,KF,X1,XF,RM1,RMF,RP1,RPF)
      IF(J.GT.1) GO TO 22
      IF(CSIGNF(KC).LT.0.) GO TO 22
      RP1=RP2
 22   CONTINUE
C  NO NEED FOR RE-AVERAGING SINCE IT INTRODUCES ONLY HIGHER ORDER
C  CHANGES INTO THE ITERATION SCHEME.
C  INTEGRATE THE CHARACTERISTIC EQUATIONS FOR RM4,RP4 AT X4,Y4.
      RM40=RM4
      RP40=RP4
      IF(CSIGNF(KC).LT.0.) GO TO 21
      RM4=RM1+DELTA*DSIN(TETA14)*D14/(M14*Y14)
      RP4=RP2+DELTA*DSIN(TETA24)*D24/(M24*Y24)
      GO TO 20
 21   CONTINUE
      RM4=RM2+DELTA*DSIN(TETA24)*D24/(M24*Y24)
      RP4=RP1+DELTA*DSIN(TETA14)*D14/(M14*Y14)
 20   CONTINUE
C  CONVERGENCE TEST
      EPS=(DABS(X4-X40)+DABS(X1-X10))/DY+DABS(RM4-RM40)+DABS(RP4-RP40)
      IF(ITER.GT.ITER0) GO TO 10
      IF(EPS.GT.EPSIL) GO TO 1
      CSIGNN(KC)=CSIGNF(KC)
      IF(X4.GT.0.) GO TO 30
      RMSAVE=RM4
      RM4=RP4+TETSYM
      RP4=RM4-TETSYM
      CSIGNN(KC)=-1.D0
 30   CONTINUE
      RMCARN(KC)=RM4
      RPCARN(KC)=RP4
      CALL RFUNC(RM4,RP4,M4,MU4,TETA4)
      TCHARN(KC)=TETA4
      XCHARN(KC)=DABS(X4)
      YCHARN(KC)=Y4
      MUCARN(KC)=MU4
      MCHARN(KC)=M4
 100  CONTINUE
      RETURN
 10   CONTINUE
      PRINT 11,KC,KCHAR0,IM,KF,ITER,ITER0,EPS,EPSIL,X1,X2,X4,M14,M24
 11   FORMAT(1X,'SUBR. SEMINV. KC,KCHAR0,IM,KF,ITER,ITER0=',6I5/
     1       1X,'EPS,EPSIL,X1,X2,X4,M14,M24=',7D14.6/)
      CALL FIN(711)
```

```
      RETURN
      END
```

RFUNC

```
      SUBROUTINE RFUNC(RM,RP,M,MU,TETA)
C  SUBROUTINE NUMBER  8
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1             TETAF(101),BF(101),
     2             XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3             TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
C
C  COMPUTE M,MU,TETA AT A POINT, AS FUNCTION OF RIEMANN INVAR. RM,RP.
      TETA=(RM-RP)/2.D0+TETLIM
      NU  =(RM+RP)/2.D0
C  NU=NU0-(G5*ARCTAN(G5*Q)-ARCTAN(Q)), WHERE Q=(M**2-1)**(-1/2)
C  FIND Q(NU), AND HENCE M(NU), THROUGH NEWTON RAPHSON ITERATIONS.
      Q=-(NU-NU0)/(G4-1.D0)
      IF(Q.LE.0.) CALL FIN(801)
      ITER=0
 1    ITER=ITER+1
      QF=Q
      DNUDT=-(G4-1.D0)/((1.D0+G4*Q**2)*(1.D0+Q**2))
      DNU=NU-(NU0-(G5*DATAN(G5*Q)-DATAN(Q)))
      Q=Q+DNU/DNUDT
      IF(Q.LE.0.) CALL FIN(811)
      EPS=DABS(Q-QF)/Q
      IF(ITER.GT.ITER0) GO TO 10
      IF(EPS.GT.EPSIL*1.D-3) GO TO 1
      M=DSQRT(1.D0+1.D0/Q**2)
      MU=DARSIN(1.D0/M)
      RETURN
 10   CONTINUE
      CALL FIN(810)
      RETURN
      END
```

INTERP

```
      SUBROUTINE INTERP(JNEW,I,KGRID,X,XVEC,RM,RMVEC,RP,RPVEC)
C  SUBROUTINE NUMBER  9
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1             TETAF(101),BF(101),
     2             XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3             TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      DIMENSION XVEC(1),RMVEC(1),RPVEC(1)
C
C  FIND I SUCH THAT XVEC(I).LE.X.AND.XVEC(I+1).GE.X
C  FIND RM,RP BY LINEAR INTERPOLATION.
C  NOTE THAT X MAY BE NEGATIVE.
      IF(DABS(X).LE.XVEC(KGRID)) GO TO 901
      PRINT 900,X,KGRID,XVEC(KGRID)
FORMAT(/1X,D15.7,I10,4X,D15.7/)
      CALL FIN(900)
  901 CONTINUE
      KG2=2*KGRID
```

```
      I0=MIN0(I,KGRID-2)
      ICOUNT=0
 1    I=I0
      SIGN1=1.D0
      IF(I.GE.1) GO TO 10
      SIGN1=-1.D0
      I=-I+2
 10   CONTINUE
      IF(I.GT.KGRID) CALL FIN(901)
      XX1=SIGN1*XVEC(I)
      I1=I
      IF(XX1.LE.X) GO TO 11
      I0=I0-1
      ICOUNT=ICOUNT+1
      IF(ICOUNT.GT.KG2) CALL FIN(911)
      GO TO 1
 11   CONTINUE
      I=I0+1
      SIGN2=1.D0
      IF(I.GE.1) GO TO 12
      SIGN2=-1.D0
      I=-I+2
 12   CONTINUE
      IF(I.GT.KGRID) CALL FIN(912)
      XX2=SIGN2*XVEC(I)
      I2=I
      IF(XX2.GE.X) GO TO 13
      I0=I0+1
      ICOUNT=ICOUNT+1
      IF(ICOUNT.GT.KG2) CALL FIN(913)
      GO TO 1
 13   CONTINUE
      F1=(XX2-X)/(XX2-XX1)
      F2=1.D0-F1
      IF(F1.LT.0.) CALL FIN(991)
      IF(F2.LT.0.) CALL FIN(992)
      RM1=RMF(I1)
      RP1=RPF(I1)
      RM2=RMF(I2)
      RP2=RPF(I2)
      IF(SIGN1.LT.0.) RM1=RPF(I1)+TETSYM
      IF(SIGN1.LT.0.) RP1=RMF(I1)-TETSYM
      IF(SIGN2.LT.0.) RM2=RPF(I2)+TETSYM
      IF(SIGN2.LT.0.) RP2=RMF(I2)-TETSYM
      RM=F1*RM1+F2*RM2
      RP=F1*RP1+F2*RP2
      RETURN
      END                                              INTERX
      SUBROUTINE INTERX(JNEW,I1,VAR0,VAR,KGRID,X0,XVEC)
C  SUBROUTINE NUMBER  10
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1            TETAF(101),BF(101),
     2            XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3            TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1            G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1           STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2           TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1           KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      DIMENSION VAR(1),XVEC(1)
C  FIND X0 AND I1 SUCH THAT XVEC(I1)<X0<XVEC(I1+1), AND X0 CORRESPONDS
C  TO THE LOCATION AT WHICH VAR0 IS A LINEAR INTERPOLATION OF VAR(I).
      X0=1.D23
      IFIRST=I1
      IF(I1.GT.0) GO TO 10
      IFIRST=KGRID-IABS(I1)+2
```

```
 10    CONTINUE
       DO 1 II=IFIRST,KGRID
       I=II
       IF(I1.GT.0) GO TO 11
       I=KGRID-II+2
 11    CONTINUE
       IF(I.LE.0) CALL FIN(1001)
       IF(I.GT.KGRID) CALL FIN(1002)
       IF(I.EQ.1) GO TO 1
       IF((VAR(I)-VAR0)*(VAR(I-1)-VAR0).GT.0.) GO TO 1
       IF(VAR(I).EQ.VAR(I-1)) GO TO 1
       F1=(VAR(I)-VAR0)/(VAR(I)-VAR(I-1))
       F2=1.D0-F1
       IF(F1.LT.0.) CALL FIN(1011)
       IF(F2.LT.0.) CALL FIN(1012)
       X0=F1*XVEC(I-1)+F2*XVEC(I)
       I1=I-1
       GO TO 2
 1     CONTINUE
 2     CONTINUE
       RETURN
       END
```

BREAK

```
       SUBROUTINE BREAK
C  SUBROUTINE NUMBER  11
       IMPLICIT REAL*8(A-H,L-Z,$)
       REAL MB,MX,MY
       COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
      1             TETAF(101),BF(101),
      2             XN(101),RMN(101),RPN(101),MN(101),MUN(101),
      3             TETAN(101),BN(101),XTEMP(101)
       COMMON/THICKY/XTH(1002),TH(1002)
       COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
      1             G16,G17,G18,G19,G20
       COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
      1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
      2            TETSYM,TETLIM,DDY,DYMAX
       COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
       COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
      1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
       COMMON /ROW/YF,YN,DXF,DXN
C
C  COMPUTE THE BREAKDOWN PARAMETER AT (I-1/2,K-1/2).  STORE IN BN(I).
       YB=0.5D0*(YF+YN)
       DYY=DY
       IM=2
       DO 1 I=2,KN
       X1=XN(I-1)
       X2=XN(I)
       DXX=X2-X1
       IF(X2.GT.XF(KF)) GO TO 2
       CALL INTERP(0,IM,KF,X1,XF,RM1,RMF,RP1,RPF)
       CALL INTERP(0,IM,KF,X2,XF,RM2,RMF,RP2,RPF)
       CALL RFUNC(RM1,RP1,M1,MU1,TETA1)
       CALL RFUNC(RM2,RP2,M2,MU2,TETA2)
       MX=0.5D0*((MN(I)-MN(I-1))+(M2-M1))/DXX
       MY=0.5D0*((MN(I)-M2)+(MN(I-1)-M1))/DYY
       MB=0.25D0*(MN(I-1)+MN(I)+M1+M2)
       TETAB=0.25D0*(TETAN(I-1)+TETAN(I)+TETA1+TETA2)
       GOREM=MB**2*(1.D0+G1*MB**2)**(G6-1.D0)
       GRAD=MX*DCOS(TETAB)+MY*DSIN(TETAB)
       B=G20*GOREM*GRAD
       GO TO 3
 2     B=1.D22
 3     BN(I)=B
 1     CONTINUE
       BN(1)=BN(2)
       RETURN
       END
```

OPACY

```
       SUBROUTINE OPACY
C  SUBROUTINE NUMBER  12
       IMPLICIT REAL*8(A-H,L-Z,$)
       COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
```

```
     1              TETAF(101),BF(101),
     2              XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3              TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1              G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1             STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2             TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1              KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
C
C  COMPUTE THE MOLECULAR THICKNESS AT END POINTS OF EACH ROW.
      IM=2
      XTH(J)=XF(KF)
      TH(J)=0.
      DTH0=N0*SIGMA*DY
      IF(J.EQ.1) GO TO 11
      J1=J-1
      DO 1 JJ=1,J1
      XX1=XTH(JJ)
      CALL INTERP(0,IM,KF,XX1,XF,RM1,RMF,RP1,RPF)
      CALL RFUNC(RM1,RP1,M1,MU1,TETA1)
      GOREM=1.D0+G1*M1**2
      DTH=DTH0/GOREM**G6
      TH(JJ)=TH(JJ)+DTH
 1    CONTINUE
 11   CONTINUE
      RETURN                                     PLUMES
      END
```

```
      SUBROUTINE PLUMES
C  SUBROUTINE NUMBER  13
      IMPLICIT REAL*8(A-H,L-Z,$)
      REAL*4 XPL,YPL
      COMMON /PLUME/XPL(1002,10),YPL(1002)
      COMMON /IPLUME/KPL,ITYPL(10)
      DIMENSION VPL(92)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1              TETAF(101),BF(101),
     2              XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3              TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      REAL*4 YXI,XI,XIPM,XIGRP,XIAPP,XIF
      COMMON /THICKX/YXI(20),XI(101,20),XIPM(101,20),XIGRP(101,20)
     1               ,XIAPP(101,20),XIF(101,20)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1              G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1             STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2             TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1              KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1                RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2                TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3                CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4                MCHARI(92)
      COMMON /ICHARA/KCHARP,KCHARM,KCHARO
C  COMPUTE SPECIAL POINTS AT Y=YN, AND STORE THEM AS
C        (XPL(J,IPL),YPL(J)=YN).
C  J IS THE MARCHING INDEX OF YN.
C  IPL=1,2,...,KPL IS THE "PLUME" INDEX.  PRESENTLY  KPL.LE.5
C  VPL(IPL) IS A VALUE DEFINING THE "PLUME" CURVE.
C  ITYPL(IPL) IS THE TYPE OF CURVE.  IT DEFINES CURVES AS FOLLOWS:
C  ITYPL(IPL)=0  DO NOTHING
C  ITYPL(IPL)=1  REAL PLUME.  IT IS THE BREAKDOWN SURFACE, DEFINED
C                BY A CONSTANT VALUE OF THE BREAKDOWN PARAMETER B.
C                SET  VPL(IPL)=B.
```

```
C  ITYPL(IPL)=2  CONSTANT MACH-NUMBER LINE.  VPL(IPL)=M.
C  ITYPL(IPL)=3  A SINGLE STREAMLINE.  VPL(IPL) IS SET TO THE EXIT
C                X-COORDINATE OF THAT STREAMLINE.
C  ITYPL(IPL)=4  A SINGLE C+ CHARACTERISTIC LINE STARTING AT THE CORNER.
C                VPL(IPL) IS SET TO THE INDEX KC OF THAT CHARACTERISTIC
C                LINE.
C  ITYPL(IPL)=5  A CONSTANT LATERAL (X) OPACITY LINE.  VPL(IPL) IS SET
C                TO THE VALUE OF THE (CONSTANT) OPACITY.
C
C  DEFINE ITYPL(IPL) AND VPL(IPL)
      KPL=10
      IF(KPL.GT.10) CALL FIN(1301)
      DO 2000 IPL=1,KPL
      GO TO (2001,2002,2003,2004,2005,2006,2007,2008,2009,2010),IPL
2001  ITYPL(IPL)=4
      VPL(IPL)=1
      GO TO 2000
2002  ITYPL(IPL)=4
      VPL(IPL)=KCHARP
      GO TO 2000
2003  ITYPL(IPL)=4
      VPL(IPL)=19
      GO TO 2000
2004  ITYPL(IPL)=4
      VPL(IPL)=31
      GO TO 2000
2005  ITYPL(IPL)=4
      VPL(IPL)=47
      GO TO 2000
2006  ITYPL(IPL)=4
      VPL(IPL)=55
      GO TO 2000
2007  ITYPL(IPL)=1
      VPL(IPL)=0.02D0
      GO TO 2000
2008  ITYPL(IPL)=1
      VPL(IPL)=0.03D0
      GO TO 2000
2009  ITYPL(IPL)=1
      VPL(IPL)=0.05D0
      GO TO 2000
2010  ITYPL(IPL)=1
      VPL(IPL)=0.08D0
      GO TO 2000
 2000 CONTINUE
C  COMPUTE "PLUME" POINTS AT Y=YN
      DO 1000 IPL=1,KPL
      ITYP=ITYPL(IPL)
      IF(ITYP.EQ.0) GO TO 1000
      GO TO (1,2,3,4,5), ITYP
 1    CONTINUE
C  BREAKDOWN SURFACE PLUME.
C  NOTE THAT DUE TO DIFFERENCE-CENTERING OF GRADIENTS, THE ACCURATE
C  Y-COORDINATE IS 0.5*(YF+YN), RATHER THAN YN.  IT CAN BE ADJUSTED
C  IN THE PLOTTING CODE.
      B0=VPL(IPL)
      XTEMP(1)=XN(1)
      DO 11 I=2,KN
      XTEMP(I)=0.5D0*(XN(I)+XN(I-1))
 11   CONTINUE
      I=2
      CALL INTERX(1,I,B0,BN,KN,XB0,XTEMP)
      XPL(J,IPL)=XB0
      GO TO 1001
 2    CONTINUE
C  FIND BY INTERPOLATION THE X-COORDINATE WHERE M=MPL.
      IF(J.GT.1) GO TO 200
      XPL(J,IPL)=XC
      GO TO 1001
 200  CONTINUE
      MPL=VPL(IPL)
      I=-KN
```

```
      CALL INTERX(1,I,MPL,MN,KN,XM0,XN)
      XPL(J,IPL)=XM0
      GO TO 1001
 3    CONTINUE
C  STREAMLINE INTERPOLATION.
      IF(J.GT.1) GO TO 300
      XPL(J,IPL)=VPL(IPL)
      GO TO 1001
 300  CONTINUE
      XSF=XPL(J-1,IPL)
      ISF=2
      ISN=2
      CALL INTERP(0,ISF,KF,XSF,XF,RMSF,RMF,RPSF,RPF)
      CALL RFUNC(RMSF,RPSF,MSF,MUSF,TETASF)
      XSN=XSF+DY*DTAN(PAI2-TETASF)
      ITER=1
 301  ITER=ITER+1
      CALL INTERP(1,ISN,KN,XSN,XN,RMSN,RMN,RPSN,RPN)
      CALL RFUNC(RMSN,RPSN,MSN,MUSN,TETASN)
      TETAAV=0.5D0*(TETASF+TETASN)
      XSN=XSF+DY*DTAN(PAI2-TETAAV)
      IF(ITER.LT.ITER0+2) GO TO 301
      XPL(J,IPL)=XSN
      GO TO 1001
 4    CONTINUE
C  CHARACTERISTIC LINE.
      KC=IDINT(VPL(IPL)+1.D-5)
      IF(J.GT.1) GO TO 41
      XPL(J,IPL)=XCHARF(KC)
      GO TO 1001
 41   CONTINUE
      XPL(J,IPL)=XCHARN(KC)
      IF(CSIGNN(KC).EQ.0.) XPL(J,IPL)=1.E33
      GO TO 1001
 5    CONTINUE
C  CONSTANT LATERAL (X) OPACITY
      CALL OPACX
      XIC=VPL(IPL)
      DO 51 II=2,KF
      I1=KF-II+1
      I2=I1+1
      XI1=XI(I1,JXI)
      XI2=XI(I2,JXI)
      IF((XIC-XI1)*(XIC-XI2).GT.0.) GO TO 51
      F2=(XI2-XIC)/(XI2-XI1)
      F1=1.D0-F2
      IF(F1.LT.0.) CALL FIN(1351)
      IF(F2.LT.0.) CALL FIN(1352)
      XIFC=F2*XF(I1)+F1*XF(I2)
      GO TO 52
 51   CONTINUE
      XIFC=1.D30
 52   CONTINUE
      XPL(J,IPL)=XIFC
      GO TO 1001
 1001 CONTINUE
      IF(J.GT.1) GO TO 1002
      YPL(J)=YC
      GO TO 1000
 1002 CONTINUE
      YPL(J)=YN
 1000 CONTINUE
      RETURN
      END                                          GRIDN
      SUBROUTINE GRIDN
C  SUBROUTINE NUMBER  14
      IMPLICIT REAL*8(A-H,L-Z,$)
      REAL*4 XPL,YPL
      COMMON /PLUME/XPL(1002,10),YPL(1002)
      COMMON /IPLUME/KPL,ITYPL(10)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1             TETAF(101),BF(101),
```

```
     2              XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3              TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1              G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1             STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2             TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1              KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1                RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2                TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3                CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4                MCHARI(92)
      COMMON /ICHARA/KCHARP,KCHARM,KCHAR0
C  DIVIDE LINE Y=YN INTO KN-1 INTERVALS.
C  THE X-GRID IS NON-UNIFORMLY DEFINED AS FOLLOWS:
C  (1) (XCHARN(I),YCHARN(I)), (XCHARF(I),YCHARF(I)), I=1,2,...,KCHARP,
C      DENOTE NEW AND OLD (FORMER) CHARACTERISTIC (C+) POINTS.  LET I=1
C      AND I=KCHARP CORRESPOND TO THE LEADING AND BOUNDARY
C      CHARACTERISTICS (C+).
C  (2) THE GRID CONSISTS OF TWO SEGMENTS.  THE SO-CALLED FLAT SEGMENT
C      IS BETWEEN X=0 AND X=XLEAD=XCHARN(KCLEAD).  THE SECOND IS THE
C      FAN SEGMENT.  IT IS FROM XLEAD TO XBOUND=XCHARN(KCHARP).
C  (3) THE FAN SEGMENT IS INITIALLY DIVIDED INTO FRACG*(KF0-1) INTERVALS
C      DEFINED BY THE FAMILY OF C+ CHARACTERISTIC LINES MCHARI(1) TO
C      MCHARI(KCHARP).
C  (4) THE FLAT SEGMENT IS DIVIDED INTO (1-FRACG)*(KF0-1) EQUAL
C      INTERVALS, AS LONG AS THEY ARE NOT SMALLER THAN THE AVERAGE
C      FAN INTERVAL.  WHEN THEY ARE, THEIR NUMBER IS REDUCED, BUT NOT
C      BELOW THREE.
C  (5) KCLEAD IS INITIALLY 1.  IT IS UPDATED SO THAT THE FLAT SEGMENT
C      IS AT LEAST TWICE THE AVERAGE FAN INTERVAL.
      ILEADF=ILEAD
      KCLEAD=0
      DO 1 KC=1,KCHARP
      IF(CSIGNN(KC).LT.0.) GO TO 1
      KCLEAD=KC
      KFAN=KCHARP-KCLEAD
      XLEAD=XCHARN(KCLEAD)
      XBOUND=XCHARN(KCHARP)
      DX1=(XBOUND-XLEAD)/DFLOAT(KFAN)
      IF(XLEAD/DX1.GT.2.D0) GO TO 11
    1 CONTINUE
   11 CONTINUE
      IF(KCLEAD.EQ.     0) CALL FIN(1401)
      IF(KCLEAD.EQ.KCHARP) CALL FIN(1402)
      ILEAD=IDINT(XLEAD/DX1)+2
      IF(ILEAD+KFAN.GT.KF0) ILEAD=KF0-KFAN
      ILEAD1=ILEAD-1
      KN=ILEAD+KFAN
      IF(KN.GT.KF0) CALL FIN(1411)
      DX=XLEAD/DFLOAT(ILEAD1)
      XN(1)=0.
      DO 2 I=1,ILEAD1
      XN(I)=XN(1)+DX*DFLOAT(I-1)
    2 CONTINUE
      DO 3 I=ILEAD,KN
      XN(I)=XCHARN(KCLEAD+I-ILEAD)
    3 CONTINUE
      RETURN
      END
```

YSTEP

```
      SUBROUTINE YSTEP
C  SUBROUTINE NUMBER  15
      IMPLICIT REAL*8(A-H,L-Z,$)
      REAL*4 XPL,YPL
      COMMON /PLUME/XPL(1002,10),YPL(1002)
      COMMON /IPLUME/KPL,ITYPL(10)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
```

```
     1              TETAF(101),BF(101),
     2              XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3              TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1              G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1             STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2             TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1              KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1                RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2                TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3                CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4                MCHARI(92)
      COMMON /ICHARA/KCHARP,KCHARM,KCHAR0
C  COMPUTE NEXT Y-STEP.
C  DYNEXT IS DEFINED AS THE MINIMAL "TRIANGULATION" Y-STEP DYT, OBTAINED
C  BY FORWARD INTERSECTION OF C-,C+ CHARACTERISTICS FROM ADJACENT GRID
C  POINTS X1,X2.
      DYMIN=1.D40
      DO 1 I=3,KF
      X1=XF(I-1)
      X2=XF(I)
      DX=X2-X1
      TP1=DTAN(TETAF(I-1)-MUF(I-1))
      TP2=DTAN(TETAF(I)+MUF(I))
      F1=-TP2/(TP1-TP2)
      IF(F1.LE.0.) PRINT 555,I,X1,X2,DX,TP1,TP2,F1
 555  FORMAT(/1X,'I,X1,X2,DX,TP1,TP2,F1=',I5,6D14.6/)
      IF(F1.LT.0.) CALL FIN(1501)
      DYT=F1*DX*TP1
      IF(DYT.LE.0.) CALL FIN(1502)
      DYMIN=DMIN1(DYMIN,STAB*DYT)
 1    CONTINUE
      DYNEXT=DYMIN
      RETURN
      END
```

MOVE

```
      SUBROUTINE MOVE
C  SUBROUTINE NUMBER  16
      IMPLICIT REAL*8(A-H,L-Z,$)
      REAL*4 XPL,YPL
      COMMON /PLUME/XPL(1002,10),YPL(1002)
      COMMON /IPLUME/KPL,ITYPL(10)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1              TETAF(101),BF(101),
     2              XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3              TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1              G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1             STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2             TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1              KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1                RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2                TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3                CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4                MCHARI(92)
      COMMON /ICHARA/KCHARP,KCHARM,KCHAR0
C  STORE NEW LINE (N) IN OLD LINE (F).
      KF=KN
      KF2=2*KF
      YF=YN
      DO 1 I=1,KN
```

```
      XF(I)=XN(I)
      RMF(I)=RMN(I)
      RPF(I)=RPN(I)
      MF(I)=MN(I)
      MUF(I)=MUN(I)
      TETAF(I)=TETAN(I)
      BF(I)=BN(I)
  1   CONTINUE
      DO 2 KC=1,KCHAR0
      IF(CSIGNN(KC).EQ.0.) GO TO 2
      XCHARF(KC)=XCHARN(KC)
      YCHARF(KC)=YCHARN(KC)
      RMCARF(KC)=RMCARN(KC)
      RPCARF(KC)=RPCARN(KC)
      TCHARF(KC)=TCHARN(KC)
      MUCARF(KC)=MUCARN(KC)
      MCHARF(KC)=MCHARN(KC)
      CSIGNF(KC)=CSIGNN(KC)
  2   CONTINUE
      RETURN
      END
```

OPACX

```
      SUBROUTINE OPACX
C  SUBROUTINE NUMBER  17
      IMPLICIT REAL*8(A-H,L-Z,$)
      REAL*4 XPL,YPL
      COMMON /PLUME/XPL(1002,10),YPL(1002)
      COMMON /IPLUME/KPL,ITYPL(10)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1            TETAF(101),BF(101),
     2            XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3            TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      REAL*4 YXI,XI,XIPM,XIGRP,XIAPP,XIF
      COMMON /THICKX/YXI(20),XI(101,20),XIPM(101,20),XIGRP(101,20)
     1              ,XIAPP(101,20),XIF(101,20)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1            G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1           STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2           TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1              RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2              TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3              CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4              MCHARI(92)
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1            KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
C  COMPUTE X-OPACITY.
C  BEGIN FROM LIMITING CHARACTERISTIC OF AN ASSUMED P.M. FAN.
C  XI0  --  THE THICKNESS BETWEEN THE LIMITING CHARACTERISTIC AND THE
C           BOUNDARY CHARACTERISTIC OF THE NUMERICAL COMPUTATION.
      DO 12 I=1,KF0
      XIF(I,JXI)=XF(I)
      XI(I,JXI)=0.
      XIPM(I,JXI)=0.
      XIGRP(I,JXI)=0.
      XIAPP(I,JXI)=0.
 12   CONTINUE
      IF(J.EQ.1) GO TO 1000
      PSILIM=TETLIM
      XLIM=XC+(YF-YC)/DTAN(PSILIM)
      XBOUND=XF(KF)
      KPM=10
      DX=(XLIM-XBOUND)/DFLOAT(KPM)
      SUM=0.
      DO 1 I=1,KPM
      X1=XBOUND+DFLOAT(I-1)*DX
      X2=X1+DX
      PS1=PAI2-DATAN((X1-XC)/(YF-YC))
      PS2=PAI2-DATAN((X2-XC)/(YF-YC))
```

```
      Q1=(PS1-PSILIM)/G5
      Q2=(PS2-PSILIM)/G5
      IF(I.EQ.KPM) Q2=1.D-10
      IF(Q2.LT.0.) CALL FIN(1701)
      F1=G11*(DSIN(Q1))**(2.D0/(G-1.D0))
      F2=G11*(DSIN(Q2))**(2.D0/(G-1.D0))
      SUM=SUM+DX*(F1+F2)/2.D0
 1    CONTINUE
      XI0=SUM*(N0*SIGMA)
C  RE-EVALUATE XI0 FOR A RING-JET.
      IF(DELTA.EQ.0.) GO TO 14
      M=MFIN
      CALL MFUNC(M,F,ETA,TETA)
      PSI=TETA+DARSIN(1.D0/M)
      GOREM=1.D0+G1*M**2
      GOR=M**2-1.D0
      CALL HINTER(M,HM)
      DELTOB=0.5D0*DSQRT(GOR)*(1.D0/(MEXIT*ETA)+DSIN(TETA)/M)/DSIN(PSI)
     1   +G15*HM/2.D0
      EVER=SIGMA*N0*YC/(M*DSIN(TETA)*DSIN(PSI)*GOREM**G6)
      GGG=2.D0-DELTOB*(G+1.D0)/2.D0
      IF(DABS(GGG).GT.1.D-10) GO TO 15
      PRINT 16, DELTOB,G,GGG
 16   FORMAT(/1X,'FROM OPACX. GGG NEARLY ZERO. EXPRESSION FOR XI0 IS',
     1         1X,'SINGULAR.  DELTOB,G,GGG=',3D12.4/)
      CALL FIN(1715)
 15   CONTINUE
      EVER=EVER/GGG
      XI0=EVER*((YF/YC)**GGG-1.D0)/(YF/YC)
 14   CONTINUE
      XI(KF,JXI)=XI0
      XIPM(KF,JXI)=XI0
      XIGRP(KF,JXI)=XI0
      KF1=KF-1
      DO 2 II=1,KF1
      I=KF-II+1
      X1=XF(I)
      X2=XF(I-1)
      DX=X1-X2
      F1=1.D0/(1.D0+G1*MF(I  )**2)**G6
      F2=1.D0/(1.D0+G1*MF(I-1)**2)**G6
      DTNUM=(N0*SIGMA)*DX*(F1+F2)/2.D0
      XI(I-1,JXI)=XI(I,JXI)+DTNUM
      XIPM(I-1,JXI)=1.D24
      XIGRP(I-1,JXI)=1.D24
      PS1=PAI2-DATAN((X1-XC)/(YF-YC))
      PS2=PAI2-DATAN((X2-XC)/(YF-YC))
      IF(PS2.GT.PSI1) GO TO 2
      Q1=(PS1-PSILIM)/G5
      Q2=(PS2-PSILIM)/G5
      IF(Q1.LT.0.) CALL FIN(1711)
      F1=G11*(DSIN(Q1))**(2.D0/(G-1.D0))
      F2=G11*(DSIN(Q2))**(2.D0/(G-1.D0))
      DTPM=(N0*SIGMA)*DX*(F1+F2)/2.D0
      XIPM(I-1,JXI)=XIPM(I,JXI)+DTPM
      DIST1=DSQRT((X1-XC)**2+(YF-YC)**2)
      DIST2=DSQRT((X2-XC)**2+(YF-YC)**2)
      KC1=KCLEAD+I-ILEAD
      KC2=KC1-1
      IF(KC2.LT.KCLEAD) GO TO 21
      M1=MCHARI(KC1)
      M2=MCHARI(KC2)
      CALL MATCH(I  ,M1,MG1,MOBI1,MABI1)
      CALL MATCH(I-1,M2,MG2,MOBI2,MABI2)
      F1=1.D0/(1.D0+G1*MG1**2)**G6
      F2=1.D0/(1.D0+G1*MG2**2)**G6
      DTGRP=(N0*SIGMA)*DX*(F1+F2)/2.D0
      XIGRP(I-1,JXI)=XIGRP(I,JXI)+DTGRP
 21   CONTINUE
 2    CONTINUE
C  APPROXIMATE THICKNESS XIAPP(I,JXI). BASED ON CLOSED-FORM INTEGRATION.
      DO 3 I=1,KF
```

```
      XIAPP(I,JXI)=1.D24
      KC=KCLEAD+(I-ILEAD)
      IF(DELTA.EQ.0.) GO TO 3
      IF(KC.LT.KCLEAD) GO TO 3
      IF(XF(I).LT.XCHARF(1)) GO TO 3
      M=MCHARI(KC)
      CALL MFUNC(M,F,ETA,TETA)
      PSI=TETA+DARSIN(1.D0/M)
      GOREM=1.D0+G1*M**2
      GOR=M**2-1.D0
      CALL HINTER(M,HM)
      DELTOB=0.5D0*DSQRT(GOR)*(1.D0/(MEXIT*ETA)+DSIN(TETA)/M)/DSIN(PSI)
     1   +G15*HM/2.D0
      EVER=SIGMA*N0*YC/(M*DSIN(TETA)*DSIN(PSI)*GOREM**G6)
      GGG=2.D0-DELTOB*(G+1.D0)/2.D0
      IF(DABS(GGG).GT.1.D-10) GO TO 25
      PRINT 26, I,KC,M,DELTOB,G,GGG
   26 FORMAT(/1X,'FROM OPACX. GGG NEARLY ZERO. EXPRESSION FOR XI0 IS',
     1        1X,'SINGULAR.  I,KC,M=',I5,D12.4/
     2        1X,'DELTOB,G,GGG=',3D12.4/)
      CALL FIN(1725)
   25 CONTINUE
      EVER=EVER/GGG
      XIAPP(I,JXI)=EVER*((YF/YC)**GGG-1.D0)/(YF/YC)
    3 CONTINUE
 1000 CONTINUE
      RETURN
      END
```

LOADC

```
      SUBROUTINE LOADC
C  SUBROUTINE NUMBER  18
      IMPLICIT REAL*8(A-H,L-Z,$)
      REAL*4 XPL,YPL
      COMMON /PLUME/XPL(1002,10),YPL(1002)
      COMMON /IPLUME/KPL,ITYPL(10)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1             TETAF(101),BF(101),
     2             XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3             TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      REAL*4 YXI,XI,XIPM,XIGRP,XIAPP,XIF
      COMMON /THICKX/YXI(20),XI(101,20),XIPM(101,20),XIGRP(101,20)
     1             ,XIAPP(101,20),XIF(101,20)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1               RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2               TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3               CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4               MCHARI(92)
      COMMON /ICHARA/KCHARP,KCHARM,KCHAR0
C  LOAD FLOW VARIABLES OF GRID POINTS IN THE FAN SEGMENT FROM THE
C  SEMI-INVERSE INTEGRATION (IN SUBR. SEMINV).  NOTE THAT GRID POINTS
C  XN(I) WERE ALREADY DETERMINED IN SUBR. GRIDN.
      DO 1 I=ILEAD,KN
      KC=KCLEAD+I-ILEAD
      IF(KC.GT.KCHARP) CALL FIN(1801)
      RMN(I)=RMCARN(KC)
      RPN(I)=RPCARN(KC)
      MN(I)=MCHARN(KC)
      MUN(I)=MUCARN(KC)
      TETAN(I)=TCHARN(KC)
    1 CONTINUE
      RETURN
      END
```

NUFUNC

```
      DOUBLE PRECISION FUNCTION NUFUNC(M)
```

```
C  SUBROUTINE NUMBER  19
      IMPLICIT REAL*8(A-H,L-Z,$)
      REAL*4 XPL,YPL
      COMMON /PLUME/XPL(1002,10),YPL(1002)
      COMMON /IPLUME/KPL,ITYPL(10)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1             TETAF(101),BF(101),
     2             XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3             TETAN(101),BN(101),XTEMP(101)
      COMMON/THICKY/XTH(1002),TH(1002)
      REAL*4 YXI,XI,XIPM,XIGRP,XIAPP,XIF
      COMMON /THICKX/YXI(20),XI(101,20),XIPM(101,20),XIGRP(101,20)
     1              ,XIAPP(101,20),XIF(101,20)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            TETSYM,TETLIM,DDY,DYMAX
      COMMON /STAG/RHO0,N0,P0,T0,A0,MDOT1
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /CHARAC/XCHARF(92),YCHARF(92),XCHARN(92),YCHARN(92),
     1               RMCARF(92),RPCARF(92),RMCARN(92),RPCARN(92),
     2               TCHARF(92),TCHARN(92),MUCARF(92),MUCARN(92),
     3               CSIGNN(92),CSIGNF(92),MCHARN(92),MCHARF(92),
     4               MCHARI(92)
      COMMON /ICHARA/KCHARP,KCHARM,KCHAR0
C  COMPUTE NU AS FUNCTION OF MACH NUMBER M.  NOTE THAT THE P.M.
C  DEFINITION OF NU HAS BEEN MODIFIED BY ADDING A CONSTANT.  THE USUAL
C  CHOICE OF THE CONSTANT IS SUCH THAT NU=0 FOR INFINITE M.
      Q=1.D0/DSQRT(M**2-1.D0)
      NUFUNC=NU0-(G5*DATAN(G5*Q)-DATAN(Q))
      RETURN
      END
```

HMSET

```
      SUBROUTINE HMSET
C  SUBROUTINE NUMBER  20
      IMPLICIT REAL*8(A-H,L-Z,$)
      REAL*8 KAPA0B
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            TETSYM,TETLIM,DDY,DYMAX
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /GRP/DMINV,MHINV(101),HMV(101)
      COMMON /IGRP/KHM
C  A ROUTINE FOR THE C+ DERIVATIVE DUE TO RING SYMMETRY (GRP).
      KHM=51
      IF(KHM.GT.101) CALL FIN(2001)
      MINV0=1.D0/MEXIT
      DMINV=MINV0/DFLOAT(KHM-1)
      M=MEXIT
      SUM=0.
      KHM1=KHM-1
      DO 1 I=1,KHM1
      MF=M
      MHINV(I)=MINV0-DFLOAT(I-1)*DMINV
      M=1.D0/MHINV(I)
      DM=M-MF
      M1=M-DM
      M2=M-DM/2.D0
      M3=M
      CALL MFUNC(M1,F1,ETA1,TETA1)
      CALL MFUNC(M2,F2,ETA2,TETA2)
      CALL MFUNC(M3,F3,ETA3,TETA3)
      SUM=SUM+DM*(F1+4.D0*F2+F3)/6.D0
      ETA=ETA3
      TETA=TETA3
      PSI=TETA+DARSIN(1.D0/M)
      NORM=((3.D0-G)/4.D0)*(M**2-1.D0)**0.75D0/
```

```
     1     (DSIN(PSI)*(1.D0+G1*M**2)**G14)
      HM=SUM*NORM
      HMV(I)=HM
      GOREM=1.D0+G1*M**2
      GOR=M**2-1.D0
      DELTOB=0.5D0*DSQRT(GOR)*(1.D0/(MEXIT*ETA)+DSIN(TETA)/M)/DSIN(PSI)
     1   +((G+1.D0)/(2.D0*(3.D0-G)))*HM
      EPSIOB=DELTOB/DSQRT(GOR)-DSIN(TETA)/(M*DSIN(PSI))
      KAPAOB=1.D0
      IF(DABS(PAI2-TETA).GT.1.D-6)
     1KAPAOB=DTAN(TETA)*EPSIOB
      LAMDOB=EPSIOB-DELTOB*GOREM/(GOR*DSQRT(GOR))
      PRINT 11,I,M,HM,TETA*DEG,PSI*DEG
 11   FORMAT(/1X,'                  I,M,HM,TETA,PSI=',I5,5D12.4)
      PRINT 12,DELTOB,EPSIOB*DEG,KAPAOB*DEG,LAMDOB*DEG
 12   FORMAT( 1X,'DELTOB,EPSIOB,KAPAOB,LAMDOB=',5X,5D12.4)
 1    CONTINUE
      MHINV(KHM)=0.
      HMV(KHM)=1.D0
      RETURN
      END                                           MFUNC
      SUBROUTINE MFUNC(M,F,ETA,TETA)
C  SUBROUTINE NUMBER  21
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            NUPT1,TETLIM
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /GRP/DMINV,MHINV(101),HMV(101)
C
      NU=NUFUNC(M)
      TETA=NUFUNC(MEXIT)+PAI2-NU
      GOREM=1.D0+G1*M**2
      GOR=M**2-1.D0
      F=(M**2)*(GOREM**G13)*DSIN(TETA)/GOR**1.25D0
      GOREM1=1.D0+G1*MEXIT**2
      GOR1=MEXIT**2-1.D0
      ETA=((GOREM/GOREM1)**G14)*((GOR1/GOR)**0.25D0)
      RETURN
      END
      SUBROUTINE HINTER(M,H)
C  SUBROUTINE NUMBER  22
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            TETSYM,TETLIM,DDY,DYMAX
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /GRP/DMINV,MHINV(101),HMV(101)
      COMMON /IGRP/KHM
C COMPUTE H(M) BY INTERPOLATION
      MINV=1.D0/M
      I=KHM-IDINT(MINV/DMINV-1.D-9)-1
      IF(I.GE.1.AND.I.LT.KHM) GO TO 1
      PRINT 11,I,KHM,M,MEXIT
 11   FORMAT(/1X,'I,KHM,M,MEXIT=',2I5,2D14.6/)
      CALL FIN(2201)
 1    CONTINUE
      F1=(MINV-MHINV(I+1))/DMINV
      F2=1.D0-F1
      IF(F1.LT.-1.D-9) CALL FIN(2210)
      IF(F2.LT.-1.D-9) CALL FIN(2211)
      H=F1*HMV(I)+F2*HMV(I+1)
      RETURN
      END                                           MATCH
      SUBROUTINE MATCH(I,MOB,MAB,MOBI,MABI)
C  SUBROUTINE NUMBER  23
```

```
      IMPLICIT REAL*8(A-H,L-Z,$)
      COMMON /VECS/XF(101),RMF(101),RPF(101),MF(101),MUF(101),
     1             TETAF(101),BF(101),
     2             XN(101),RMN(101),RPN(101),MN(101),MUN(101),
     3             TETAN(101),BN(101),XTEMP(101)
      COMMON /ROW/YF,YN,DXF,DXN
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1             G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1            STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2            TETSYM,TETLIM,DDY,DYMAX
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1             KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /GRP/DMINV,MHINV(101),HMV(101)
      COMMON /IGRP/KHM
C COMPUTE H(M) AND THE ALFA-DERIVATIVES
      M=MOB
      CALL MFUNC(M,F,ETA,TETA)
      PSI=TETA+DARSIN(1.D0/M)
      CALL HINTER(M,HM)
      GOREM=1.D0+G1*M**2
      GOR=M**2-1.D0
      DELTOB=0.5D0*DSQRT(GOR)*(1.D0/(MEXIT*ETA)+DSIN(TETA)/M)/DSIN(PSI)
     1   +G15*HM/2.D0
      FOB=(G7*GOREM)**G2/M
      FAB=FOB*(YF/YC)**DELTOB
      CALL AREAF(FAB,MAB)
C  COMPUTE MABI FROM THE INVERSE PROBLEM SOLUTION
      COTAV=(XF(I)-XC)/(YF-YC)
      PSI0=PAI2-DATAN(COTAV)
      EVY=YF*DLOG(YF/YC)/(YF-YC)-1.D0
      PSIN=PSI0
      DO 1 ITER=1,50
      PSI=PSIN
      M=DSQRT(1.D0+G4/DTAN((PSI-TETLIM)/G5)**2)
      M=DMAX1(M,MEXIT)
      CALL HINTER(M,HM)
      CALL MFUNC(M,F,ETA,TETA)
      GOREM=1.D0+G1*M**2
      GOR=M**2-1.D0
      DELTOB=0.5D0*DSQRT(GOR)*(1.D0/(MEXIT*ETA)+DSIN(TETA)/M)/DSIN(PSI)
     1   +G15*HM/2.D0
      EPSIOB=DELTOB/DSQRT(GOR)-DSIN(TETA)/(M*DSIN(PSI))
      LAMDOB=EPSIOB-DELTOB*GOREM/(GOR*DSQRT(GOR))
      COTN=COTAV+LAMDOB*EVY/DSIN(PSI)**2
      PSIN=PAI2-DATAN(COTN)
      DPSI=PSIN-PSI
      IF(DABS(DPSI).LT.1.D-9) GO TO 11
 1    CONTINUE
      PRINT 12,I,ITER,PSI,PSIN,DPSI,M,XF(I),YF,XC,YC
 12   FORMAT(/1X,'I,ITER,PSI,PSIN,DPSI,M,XF(I),YF,XC,YC='//
     1       1X,2I4,8D11.3/)
      CALL FIN(2301)
 11   CONTINUE
C  USING MOBI=M AS COMPUTED FROM THE INVERSE PROBLEM, FIND MABI.
      MOBI=M
      M=MOBI
      CALL MFUNC(M,F,ETA,TETA)
      PSI=TETA+DARSIN(1.D0/M)
      CALL HINTER(M,HM)
      GOREM=1.D0+G1*M**2
      GOR=M**2-1.D0
      DELTOB=0.5D0*DSQRT(GOR)*(1.D0/(MEXIT*ETA)+DSIN(TETA)/M)/DSIN(PSI)
     1   +G15*HM/2.D0
      FOB=(G7*GOREM)**G2/M
      FAB=FOB*(YF/YC)**DELTOB
      CALL AREAF(FAB,MABI)
      RETURN                                  AREAF
      END
```

```
      SUBROUTINE AREAF(F,M)
C  SUBROUTINE NUMBER  24
      IMPLICIT REAL*8(A-H,L-Z,$)
```

```
      COMMON /GAMA/G,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11,G12,G13,G14,G15,
     1            G16,G17,G18,G19,G20
      COMMON /PAR/PAI,PAI2,DEG,XC,YC,MEXIT,MFIN,YMAX,DY0,DY,DYNEXT,
     1           STAB,DELTA,PSI1,PSIF,ZETA1,SIGMA,FRACG,EPSIL,NU0,
     2           TETSYM,TETLIM,DDY,DYMAX
      COMMON /IPAR/JMAX,KF0,ITER0,KF,KN,IM,IP,J,
     1            KF2,IDEL,JDEL,JYXI,JXI,ILEAD,ILEADF,KCLEAD
      COMMON /GRP/DMINV,MHINV(101),HMV(101)
      COMMON /IGRP/KHM
C  COMPUTE MACH NUMBER M FROM AREA RATIO FUNCTION F
C  F=((2/(G+1))*(1+(G-1)*M**2))**((G+1)/(2*(G-1)))/M
C  INITIAL GUESS IS MIN
      E1=(F*MEXIT)**(1.D0/G2)/G7
      E2=(E1-1.D0)/G1
      E3=DMAX1(E2,MEXIT**2)
      MIN=DSQRT(E3)
      EMN=MIN
      DO 1 I=1,100
      EMO=EMN
      GOREM=1.D0+G1*EMO**2
      GOR=EMO**2-1.D0
      FO=(G7*GOREM)**G2/EMO
      DF=FO-F
C     PRINT 123,I,EMO,EMN,FO,F,DF,GOR,GOREM
C123  FORMAT(1X,'I,EMO,EMN,FO,F,DF,GOR,GOREM=',I5,7D12.4)
      DFDM=FO*GOR/(EMO*GOREM)
      DMN=DF/DFDM
      EMN=EMO-DMN
      EPSEM=DABS(DMN/EMN)
      IF(EPSEM.LT.1.D-10) GO TO 11
 1    CONTINUE
      CALL FIN(2401)
 11   CONTINUE
      M=EMN
      RETURN
      END
```

5. REFERENCES

[1] Falcovitz, J., "Analytic an Numerical Computation of Ring-Symmetric Spacecraft Exhaust Plumes", Naval Postgraduate School, Monterey, CA, Report NPS72-86-003CR, December 1986.

[2] Liepmann, H. W. and Roshko, A., ***Elements of Gasdynamics***, John Wiley, New York, 1957.

[3] Abramovich, S., "Gas Dynamics of Laser Exhaust External to Spacecraft", Naval Postgraduate School, Monterey, CA, Contractor Report NPS67-84-006CR, Nov. 1985.

[4] Zucrow, M. J. and Hoffman, J. D., ***Gas Dynamics***, John Wiley, New York, 1976.

[5] Bird, G. A., ***Molecular Gas Dynamics***, Clarendon Press, Oxford, 1976.

[6] Bird, G. A., "Breakdown of Continuum Flow in Free Jets and Rocket Plumes", Proceedings of 12th Symposium on Rarefied Gas Dynamics. In Volume 74, ***Progress in Astronautics and Aeronautics***, Part II, p.681, Sam S. Fisher, Editor. Published by AIAA, 1981.

[7] Falcovitz, J., "A Breakdown Surface Model for Thermal Backscattering from the Exhaust Plume of a Space-Based HF Laser", Naval Postgraduate School, Monterey, CA, Report NPS67-86-002CR, June 1986.

6. DISTRIBUTION LIST

No. of Copies

1. Defense Technical Information Center
Cameron Station
Alexandria, VA 22314 2

2. Library, Code 0142
Naval Postgraduate School
Monterey, CA 93943-5100. 2

3. Department Chairman, Code 67
Department of Aeronautics
Naval Postgraduate School
Monterey, CA 93943-5100. 1

4. Distinguished Professor Allen E. Fuhs
Space Systems Academic Group, Code 72
Naval Postgraduate School
Monterey, CA 93943-5100. 5

5. Dr. Neil Griff
SDIO/DEO
Washington, DC 20301-7100. 3

6. Mr. Bruce Pierce
SDIO/DEO
Washington, DC 20301-7100. 1

7. Dr. Joseph Falcovitz
Code 72
Naval Postgraduate School
Monterey, CA 93943-5100. 5

No. of Copies

8. Professor Max F. Platzer
Department of Aeronautics, Code 67
Naval Postgraduate School
Monterey, CA 93943-5100. 1

9. Professor Oscar Biblarz
Department of Aeronautics, Code 67
Naval Postgraduate School
Monterey, CA 93943-5100. 1

10. Professor David W. Netzer
Department of Aeronautics, Code 67
Naval Postgraduate School
Monterey, CA 93943-5100. 1

11. Research Administration Office
Code 012
Naval Postgraduate School
Monterey, CA 93943-5100. 1

12. Dr. P. Avizonis
Air Force Weapons Laboratory
Kirtland Air Force Base, NM 87117 1

13. Dr. John Lawless
Space Power Inc.
1977 Concourse Drive
San Jose, CA 95131 1

14. Dr. Mark Thornton
Boeing Aerospace Company
Post Office Box 3999
Seattle, WA 98124-2499 1

No. of Copies

15. LT. Mark Price
AFRPL
Edwards AFB, CA 93523 1

16. Mr. Arthur W. Rogers
Space Systems Division
Hughes Aircraft Co.
P. O. Box 92919, Los Angeles, CA 90009 1

17. LCOL Rick Babcock, USAF
Air Force Geophysical Laboratory
Hanscomb Field
Bedford, MA 01730 1

18. Dr. James Stark Draper
Aerodyne Research, Inc.
45 Manning Road
Billerica, MA 01821 1